Universität Hildesheim
Fachbereich III – Informations- und Kommunikationswissenschaften
Institut für Angewandte Sprachwissenschaft

Magisterarbeit
zur Erlangung des akademischen Grades eines
Magister Artium Internationales Informationsmanagement

Analyse und Einsatzpotentiale von Clustering-Verfahren zum Retrieval von Patent-Dokumenten

---

1. Gutachterin: Prof. Dr. Christa Womser-Hacker
2. Gutachter: Dr. Thomas Mandl

eingereicht von:
Joachim Pfister

Hildesheim, im November 2004

Joachim-Pfister@gmx.de

Abstract

Um dem stetigen Zuwachs der elektronisch in Datenbanken abgespeicherten Informationen wirkungsvoll zu begegnen, werden neue Werkzeuge gesucht, die den Nutzer bei Datenbankrecherchen unterstützen. In dieser Arbeit, die im Anwendungsbereich der Patentrecherche und Patentinformation angesiedelt ist, soll das automatische Gruppieren von Patentdokumenten - das so genannte Clustering - als ein Werkzeug zur Aufbereitung der Ergebnismenge einer Datenbankanfrage untersucht werden. Es werden zum einen Grundlagen der Cluster-Analyse, wie z.B. Attributtypen und Ähnlichkeits- bzw. Distanzmaße, zum anderen verschiedene Clustering-Verfahren sowie deren Vor- und Nachteile zum Clustern von Dokumenten beschrieben. Weiterhin werden Besonderheiten des Anwendungsbereichs aufgezeigt und bereits bestehende Einsatzmöglichkeiten von Clustering-Verfahren dargestellt. Im praktischen Teil dieser Arbeit werden im Anwendungsbereich Patentrecherche drei Clustering-Verfahren mittels Nutzerbewertungen miteinander verglichen, um tendenzielle Aussagen über die Eignung eines bestimmten Verfahrens abzuleiten.

Schlagworte:
Clustering, Patentdokumente, Cluster-Analyse, Dokument-Clustering, Nutzerbewertung, Vergleich, Clustering-Verfahren, Patentdatenbanken

---

The constantly growing amount of information stored in databases fosters the need for new tools, assisting the user during his or her database search. This Master's thesis relates to patent search and patent information retrieval where clustering is used as a tool to group the result set of a database query, i.e. automatically form groups of patent documents. The subjects described are principal issues of cluster analysis such as types of attributes, similarity or distance measures, different types of clustering algorithms as well as their advantages and disadvantages for clustering documents. Furthermore, the special requirements of the application area are illustrated and the existing use of clustering techniques is depicted. The thesis' practical section deals with the evaluation of three different clustering algorithms, used in the context of patent retrieval. Within this evaluation, user judgements are used to compare the three algorithms and to derive a recommendation for a specific algorithm on that basis.

Key words:
clustering, patent documents, cluster analysis, document clustering, clustering algorithms, comparison, evaluation, user judgements, patinformatics, patent databases

Inhaltsverzeichnis

Abbildungsverzeichnis viii

Tabellenverzeichnis x

Abkürzungsverzeichnis xi

1 Einleitung 1
1.1 Einleitung und Motivation 1
1.2 Aufbau der Arbeit 2

2 Grundlagen der Cluster-Analyse 4
2.1 Was ist eine Cluster-Analyse? 4
2.2 Verwandte Gebiete und Abgrenzung zur Klassifikation 5
2.3 Ablauf einer Cluster-Analyse 5

3 Anwendungsbereich Patentrecherche 8
3.1 Grundlagen des Patentwesens 8
3.1.1 Funktion von Patenten 8
3.1.2 Voraussetzungen für ein Patent 9
3.1.3 Aufbau einer Patentschrift 9
3.1.4 Klassifikation von Patentschriften 12
3.1.5 Stationen auf dem Weg zum Patent 13
3.1.6 Sprachliche und stilistische Besonderheiten von Patentschriften 13
3.2 Patentrecherche: Gründe und Infrastruktur 14
3.2.1 Die wirtschaftliche Bedeutung von Patenten 14
3.2.2 Das FIZ-Karlsruhe und seine Rolle in der Bereitstellung von Patentinformationen 16
3.2.3 Online Patentdatenbanken 17
3.2.4 Die Datenbank PATDPA 18
3.3 Zusammenfassung 19

4 Clustering im IR und im Anwendungsbereich Patentrecherche 20
4.1 Pre-Retrieval Clustering einer Kollektion 20
4.2 Post-Retrieval Clustering zur Aufbereitung von Ergebnismengen 21
4.2.1 Scatter/Gather-Ansatz 21
4.2.2 Clustern von Ergebnismengen im Web-IR 22
4.2.3 Automatisches Bezeichnen von Clustern 25

4.3 Kritik an der Darstellung von Ergebnismengen als Cluster 27
4.4 Clustering-Verfahren als Werkzeuge zur Patentanalyse und -recherche 28
4.4.1 Patinformatics und Text Mining als „Werkzeuglieferanten“ . . . 29
4.4.2 Ablauf einer Recherche und Einbindung neuer Werkzeuge zur Analyse von Patentdokumenten 30
4.5 Zusammenfassung 31

5 Auswahl und Aufbereitung der Attribute 32
5.1 Vektorraummodell und Clustering von Dokumenten 32
5.2 Attributtypen 33
5.3 Gewichtung der Terme 33
5.3.1 Gewichtung nach TF/IDF 34
5.3.2 Gewichtung nach Okapi-BM25 34
5.4 Standardisierung bzw. Normierung von Attributen 35
5.5 Zusammenfassung 37

6 Proximitätsmaße 38
6.1 Eigenschaften von Distanzmaßen 39
6.2 Minkowski-Metriken 39
6.3 Mahalanobis-Distanz 41
6.4 Ähnlichkeitsmaße bei binären Merkmalen 42
6.5 Ähnlichkeitsmaße im Vektorraummodell 43
6.6 Mutual Neighbor Distance-Verfahren 44
6.7 Weitere Proximitätsmaße 45
6.8 Zusammenfassung 45

7 Fusionierungsverfahren 46
7.1 Hierarchische Verfahren 47
7.1.1 Grundlagen hierarchischer Verfahren 47
7.1.2 Verfahren zur Bestimmung der inter-Cluster Proximität 49
7.1.2.1 Single Linkage-Verfahren 50
7.1.2.2 Complete Linkage-Verfahren 51
7.1.2.3 Average Linkage-Verfahren 52
7.1.2.4 Centroid-Verfahren 52
7.1.2.5 Median-Verfahren 53
7.1.2.6 Verfahren von Ward 53
7.2 Partitionierende Verfahren 54
7.2.1 Gütefunktionen und Refinement-Phase 56
7.2.2 K-Means – eine auf Centroiden basierende Technik 58
7.2.3 K-Medoid – eine auf Repräsentanten basierende Technik 59
7.2.4 Bisecting K-Means 59
7.3 Probabilistische Verfahren 60
7.4 Shared Nearest Neighbor Verfahren 61

7.5 Weitere Verfahren 63
7.5.1 Fuzzy-Clustering 63
7.5.2 Dichtebasierte Verfahren 64
7.5.3 Grid-basierte Verfahren 64
7.5.4 Inkrementelles Clustern 65
7.5.5 Künstliche Neuronale Netze 67
7.5.6 Evolutionäre Algorithmen 67
7.6 Zusammenfassung 69

8 Clustering-Experimente mit Patentdaten 70
8.1 Datengrundlage 70
8.1.1 Vorgehen zur Aufbereitung der Daten aus der Patentdatenbank PATDPA 70
8.1.2 Datengrundlage für die Experimente 72
8.1.2.1 Auswahl der Anfragen 72
8.1.2.2 Auswahl der Datensätze für die Experimente 73
8.2 Auswahl der Clustering-Verfahren 75
8.3 Beobachtungen in den Vorab-Versuchen 77
8.4 Durchführung der Experimente 78

9 Evaluierung 80
9.1 Cluster-Validation und mögliche Bewertungskriterien 80
9.1.1 Objektive externe Bewertungskriterien 81
9.1.1.1 F-Maß 81
9.1.1.2 Entropy und Purity 82
9.1.2 Objektive interne Bewertungskriterien 83
9.1.2.1 „Cluster cohesion“ 83
9.1.2.2 „cluster isolation“ 83
9.1.2.3 Weitere interne Bewertungskriterien 84
9.1.3 Zusammenfassung der Methoden zur Ermittlung der Cluster Validity 84
9.2 „Cluster usability“ als subjektives Bewertungskriterium 84
9.2.1 Methodik 85
9.2.2 Erhebungsplan 86
9.3 Auswertung der Experimente 87
9.3.1 Auswertung der Juroren-Beurteilungen auf Dokumentebene . . 87
9.3.2 Auswertung nach Vergabe von Schulnoten durch die Juroren . . 90
9.3.3 Auswertung der Juroren-Kommentare auf den Papier-Fragebögen 92
9.3.4 Bewertung der erzeugten Clusteranzahl 94
9.4 Schlussfolgerungen aus den Experimenten 95

10 Fazit und Ausblick 97

Literaturverzeichnis **99**

A Eingesetzte Software zur Durchführung der Clustering-Experimente **106**
A.1 CLUTO . . . 106
A.1.1 Herkunfts- und Lizenzinformationen . . . 106
A.1.2 Möglichkeiten der Software . . . 106
A.1.3 Format der Eingabedaten . . . 107
A.2 WEKA . . . 108
A.2.1 Herkunfts- und Lizenzinformationen . . . 108
A.2.2 Möglichkeiten der Software . . . 108
A.2.3 Format der Eingabedaten . . . 109
A.3 SNN-Algorithmus . . . 109
A.3.1 Herkunfts- und Lizenzinformationen . . . 109
A.3.2 Möglichkeiten der Software . . . 109
A.3.3 Format der Eingabedaten . . . 109
A.4 Autoclass-C . . . 110
A.4.1 Herkunfts- und Lizenzinformationen . . . 110
A.4.2 Möglichkeiten der Software . . . 110
A.4.3 Format der Eingabedaten . . . 110

B Im Rahmen der Magisterarbeit entwickelte Software **111**
B.1 Pre-Processing-Tool PatentPreProcess . . . 111
B.1.1 Programmeigenschaften und -fähigkeiten . . . 111
B.1.2 Konfiguration . . . 112
B.1.3 Statistiken . . . 113
B.1.4 Ablauf der Verarbeitung und Anmerkungen . . . 114
B.2 ExperimenterGUI . . . 115
B.2.1 Programmeigenschaften und -fähigkeiten . . . 115
B.2.2 Konfiguration . . . 117
B.3 Evaluierungstool ClustEv . . . 117
B.3.1 Programmeigenschaften und -fähigkeiten . . . 117
B.3.1.1 Hauptfenster . . . 118
B.3.1.2 Abgabe der Bewertungen . . . 118
B.3.1.3 Auswertung . . . 119
B.3.2 Konfiguration . . . 120

Eigenständigkeitserklärung **123**

Abbildungsverzeichnis

3.1 Deckblatt eines Patents 11
3.2 Ausschnitt aus dem Beschreibungsteil und dem Hauptanspruch einer Patentschrift 11

4.1 Darstellung einer Clustering-Lösung durch ThemeScape 29

5.1 Ausgangsdaten als Datenmatrix 32

6.1 Proximitätsmatrix 38
6.2 Dreiecksungleichung 39
6.3 City-Block-Metrik 40
6.4 Euklidische Distanz 40
6.5 Mutual Neighbor Distance 44
6.6 Mutual Neighbor Distance – Nach Veränderung des Kontexts 44

7.1 Überblick über ausgewählte Clustering-Algorithmen 46
7.2 Dendrogramm 48
7.3 Ablauf des hierarchisch-agglomerativen Clustering-Verfahrens 48
7.4 Single Linkage 50
7.5 konzentrisch angeordnete Cluster 51
7.6 Ergebnis, das mit dem Single Linkage-Verfahren entsteht. 51
7.7 Complete Linkage 51
7.8 Ergebnis, das mit dem Complete-Linkage Verfahren entsteht. 52
7.9 Average Linkage 52
7.10 Abhängigkeit des K-Means Algorithmus von der Anfangspartition . . . 56
7.11 Schritte im Erstellen einer Cluster-Lösung beim K-Means Verfahren . . 58
7.12 Beispiel für eine Mischverteilung 60
7.13 „nearest neighbor“-Graph 62
7.14 Ungewichteter „shared nearest neighbor“-Graph 62
7.15 Unregelmäßig geformte Cluster können mit dichtebasierten Verfahren ermittelt werden 64
7.16 Beispiel für eine hierarchische Strukturierung bei Grid-basierten Fusionierungsverfahren 65
7.17 Klassifikationsbaum 66
7.18 Kreuzung 68

9.1 Bewertungen der Juroren auf Dokumentebene (Absolutwerte) 89

9.2 Bewertungen der Juroren auf Dokumentebene (Normiert anhand der Anzahl erzeugter Cluster) 90
9.3 Bewertung nach Schulnoten 91
9.4 Bewertung der erzeugten Clusteranzahl 95
9.5 Bewertung nach Schulnoten - Gruppe A mit Pseudo-Lösung 96

A.1 Format der Eingabedaten 107

B.1 ExperimenterGUI 116
B.2 Darstellung eines Resultats eines Clustering-Laufes 117
B.3 Hauptfenster der Anwendung ClustEv 118
B.4 Fenster zur Bewertung einer Anfrage 119
B.5 Fenster zur Auswertung der Bewertungen 120

Tabellenverzeichnis

3.1 Sektionen der IPC 12
3.2 Beispiel für den hierarchischen Aufbau der IPC 13
3.3 Arten der Patentrecherche 16

4.1 Einzelterme im Vergleich mit LA-Termen zur Inhaltsbezeichnung für die Web-Site “Merced County“ 26

5.1 Verschiedene Skalen und ihre Eigenschaften 33
5.2 Größen zur Termgewichtung 34

6.1 Kontingenztafel für binäre Merkmale 42
6.2 Ähnlichkeitsmaße im Vektorraummodell 44

7.1 Parameter der Lance-Williams Formel für hierarchisch agglomerative Clustering-Verfahren 50
7.2 Anzahl der möglichen Partitionen von N Objekten in g Klassen 54

8.1 Statistische Werte über Anfragen an die Datenbank PATDPA zur Ermittlung der Datengrundlage für Clustering-Versuche 74
8.2 Anzahl der erzeugten Cluster 79

9.1 Aufteilung der Anfragen auf die Juroren 87
9.2 Bewertungen der Juroren auf Dokumentebene 88
9.3 Bewertungen nach Schulnoten 91
9.4 Bewertungen nach Schulnoten für alle Anfragen und Gruppen 92
9.5 Bewertung der erzeugten Clusteranzahl 94

Abkürzungsverzeichnis

AB	Abstract-Feld eines Patentdokuments
DPMA	Deutsches Patent- und Markenamt
EP, EPA	Europäisches Patentamt
FIZ	Fachinformationszentrum
IPC	International Patent Classification
IR	Information Retrieval
MainIPC	Hauptklasse eines Patentdokuments in der IPC
MCLM	Main Claim-Feld eines Patentdokuments
PATDPA	Patentdatenbank des DPMA
PATDPAFULL	Patentdatenbank des DPMA mit Volltexten
PF-D, PF-Doppel	Patentfamilien-Doppel
STN	Scientific & Technical Network
TF-IDF	Term Frequency - Inverse Document Frequency
TI	Titel-Feld eines Patentdokuments
TREC	Text Retrieval Conference
WIPO	World Intellectual Property Organization

1 Einleitung

> „An intelligent being cannot treat every object it sees as a unique entity unlike anything else in the universe. It has to put objects in categories so that it may apply its hard-won knowledge about similar objects encountered in the past, to the object at hand."
>
> *(Stephen Pinker 1997, 12)*

1.1 Einleitung und Motivation

Im vorangestellten Zitat wird das Bilden von Klassen und Kategorien als Grundprinzip beschrieben, das intelligente „Informationsverarbeiter" wie den Menschen auszeichnet. Er teilt seine Umwelt sowie die Dinge und Gegenstände, die er dort vorfindet, in Klassen und Kategorien ein. Das Verdichten vieler Einzelobjekte und -beobachtungen zu Kategorien dient dazu, Komplexität zu reduzieren und somit den „Überblick" zu bewahren.

Fortschritte in Wissenschaft und Technik führen zu unzähligen Entwicklungen, die das Leben auf zahlreichen Gebieten beeinflussen. Dabei wird eine große Menge an Informationen produziert, um durch die Dokumentation und Publikation von Ergebnissen und Vorgängen letztlich das gewonnene Wissen festzuhalten. Durch den stetigen Zuwachs an verfügbaren Informationen wird die Komplexität aller Sachverhalte erhöht, was durch ständige (Neu-)Bildung von Kategorien und Klassen kompensiert werden soll.

Ein beständiger Zuwachs an Komplexität ist in dem dieser Arbeit zu Grunde liegenden Anwendungsbereich, der Patentinformation, zu verzeichnen. Vor allem durch die wachsende wirtschaftliche Bedeutung von Patentinformationen besteht zunehmend die Notwendigkeit, einen umfassenden Überblick über vorhandene Patentschriften zu erhalten. Die vorhandenen Möglichkeiten zur Patentrecherche müssen daher beständig verfeinert und weiterentwickelt werden, um den wachsenden Anforderungen gerecht zu werden.

Im Rahmen einer Recherche nach Patentdaten erhält ein Nutzer auf seine Suchanfrage an eine Patentdatenbank bisweilen eine sehr große Trefferanzahl als Ergebnis geliefert. Diese umfangreiche Ergebnismenge muss von ihm, je nach Informationsbedürfnis, mehr oder weniger vollständig betrachtet und ausgewertet werden. An dieser Stelle knüpft die hier vorliegende Magisterarbeit an: Der Nutzer soll nicht mit

einer langen Liste von Treffern auf seine Suchanfrage konfrontiert werden. Stattdessen werden die zurückgelieferten Patentdokumente automatisch in Gruppen, den so genannten Clustern, zusammengefasst und diese werden dem Nutzer präsentiert. Dabei gilt, dass die Patentdokumente in den ermittelten Clustern möglichst ähnlich zueinander sein sollen, gleichzeitig sollen sich aber auch die Cluster möglichst gut voneinander unterscheiden.

Das automatische Zusammenführen in Cluster kann im Idealfall für den Nutzer zu einer Komplexitätsreduktion führen: Er kann auf Grund der (berechneten) Ähnlichkeit der Dokumente eines Clusters viel schneller ganze Cluster als irrelevant verwerfen und sich somit auf die verbliebenen relevanten Cluster konzentrieren. Der Hauptvorteil liegt darin, dass der Nutzer nicht mehr sämtliche Dokumente der ursprünglichen Liste betrachten muss. Das automatische Zusammenfassen von Datenobjekten in Gruppen wird auch als „clustern“ bezeichnet, das Resultat als Clustering-Lösung.

Im Zuge dieser Arbeit werden verschiedene Verfahren und Ansätze zum Clustern von Dokumenten vorgestellt. Um zu einer Aussage zu gelangen, welches Verfahren für das Clustern von Dokumenten am geeignetsten erscheint, wurde eine praktische Untersuchung durchgeführt, bei der die erzeugten Clustering-Lösungen intellektuell von Juroren bewertet wurden. Anhand dieser Bewertungen werden Schlussfolgerungen abgeleitet, um zu einer Aussage über die Eignung bestimmter Clustering-Verfahren hinsichtlich des Anwendungsbereichs Patentrecherche zu gelangen.

Das Thema dieser Arbeit entstand aus einer Kooperationsbeziehung zwischen der Inhaberin der Professur für Angewandte Informationswissenschaft an der Universität Hildesheim, Frau Prof. Dr. Womser-Hacker, und dem Fachinformationszentrum Karlsruhe (FIZ-Karlsruhe), vertreten durch Herrn Dr. Schwantner. Frau Prof. Dr. Womser-Hacker ist Mitglied im Wissenschaftlichen Beirat des FIZ-Karlsruhe und stellte anlässlich eines Magisterkolloquiums mögliche Themenbereiche für eine Abschlussarbeit in Zusammenarbeit mit dem FIZ-Karlsruhe vor, wodurch diese Arbeit letztendlich angeregt wurde.

1.2 Aufbau der Arbeit

In *Kapitel 2* werden die Grundlagen zur Cluster-Analyse, verwandte Themengebiete und der Ablauf einer Cluster-Analyse beschrieben, an dessen Vorgehensweise sich die weiteren Kapitel dieser Arbeit im Wesentlichen orientieren.

Kapitel 3 stellt den Anwendungsbereich der Patentrecherche vor, in dessen Kontext die Clustering-Verfahren eingesetzt werden. Im ersten Teil wird das Patentwesen allgemein beschrieben, um aufzuzeigen, was ein Patent ist. Im zweiten Teil werden

die Bedeutung von Patenten sowie die Infrastruktur erläutert, die im Rahmen der Patentrecherche genutzt wird.

Im darauf folgenden *Kapitel 4* werden Ansätze zum Einsatz von Clustering-Verfahren vorgestellt. Schwerpunktmäßig wird hierbei auf das Clustern von Ergebnismengen eines Retrieval-Prozesses eingegangen und die damit verbundenen Problembereiche beschrieben. Zudem werden die im Anwendungsbereich Patentrecherche bereits bestehenden Einsatzfelder von Clustering-Verfahren aufgezeigt und der Anknüpfungspunkt für Clustering-Verfahren in dieser Arbeit vorgestellt.

Mit der Auswahl und der Aufbereitung der Attribute beschäftigt sich *Kapitel 5*. Das dem Clustern von Dokumenten zu Grunde liegende Vektorraummodell wird vorgestellt sowie Ansätze zur Gewichtung und Normierung von Attributwerten.

Kapitel 6 zeigt verschiedene Verfahren und Ansätze zur Proximitätsberechnung, um Distanzen oder Ähnlichkeiten zwischen Objekten im Rahmen eines Clustering-Verfahrens zu berechnen.

In *Kapitel 7* werden Fusionierungsverfahren zur Gruppenbildung, die Clustering-Algorithmen, vorgestellt und die jeweiligen Stärken und Schwächen der Verfahren aufgezählt.

Kapitel 8 beschreibt den praktischen Teil der Magisterarbeit, nämlich die durchgeführten Clustering-Experimente mit Patentdokumenten. Neben einer Beschreibung der Datengrundlage, der Aufbereitung der Daten und der Begründung für die Auswahl der verwendeten Verfahren beinhaltet dieses Kapitel Beobachtungen, die in Vorab-Versuchen gemacht wurden und die letztlich durchgeführten Experimente mit den gewählten Parametern.

In *Kapitel 9* werden zuerst allgemein Möglichkeiten zur Evaluierung von Clustering-Lösungen vorgestellt, um anschließend die in Kapitel 8 durchgeführten Experimente auszuwerten. Die Ergebnisse des Vergleichs von drei Clustering-Verfahren werden im Anschluss präsentiert.

Kapitel 10 schließt die Arbeit mit einem Fazit und einem Ausblick ab, in dem Anregungen für weitere Untersuchungen hinsichtlich der Eignung von Clustering-Verfahren im Rahmen des Retrievals von Patent-Dokumenten gemacht werden.

Im *Anhang* befindet sich eine Art „Software-Handbuch“, das zum einen die eingesetzten Software-Tools für das Clustering beschreibt und zum anderen die im Zuge dieser Magisterarbeit entwickelten (Hilfs-)Programme zur Durchführung der Cluster-Analyse dokumentiert.

2 Grundlagen der Cluster-Analyse

In diesem Kapitel werden der Ablauf sowie die Einsatzmöglichkeiten einer Cluster-Analyse beschrieben und der grundlegende Unterschied zwischen einer Cluster-Analyse und der Klassifikation von Objekten aufgezeigt.

2.1 Was ist eine Cluster-Analyse?

Das Bilden von Kategorien oder Klassen gehört zu den grundlegenden Fähigkeiten von Menschen, um mit großen Mengen an Informationen umzugehen. Im Bereich der Wissenschaft ist die Klassifikation von Objekten ein fundamentaler Baustein, wie z.B. in der Biologie. So versuchte beispielsweise Aristoteles das Tierreich systematisch zu untergliedern, um eine so genannte Taxonomie (griech. „taxis“ = Anordnung, „nemein“ = verteilen, (Wahrig 2000, 1240)) zu erzeugen. Er ging dabei von zwei Hauptklassen aus: den Tieren mit rotem Blut und den Tieren ohne rotes Blut (Everitt et al. 2001, 1). Vor allem in der Biologie und Zoologie wurde nach numerischen Methoden gesucht, um die oft auf subjektiver Basis erstellten Taxonomien durch objektive und stabile Klassifikationsschemata zu ersetzen, die auf Grund von Berechnungen entstanden sind.

Je nach Anwendungsgebiet erhalten diese numerischen Verfahren verschiedene Bezeichnungen: „Numerische Taxonomie“ in der Biologie, „Q-Analyse“ in der Psychologie, „Segmentierung“ in der Marktforschung, und im Bereich der Künstlichen Intelligenz wird oft der Begriff „unüberwachtes Lernen“ verwendet. Im Allgemeinen wird heute eher von „Cluster-Analyse“ gesprochen, wenn Gruppen in Daten ermittelt werden sollen (vgl. Everitt et al. 2001, 4).

Ziel der Cluster-Analyse ist es, Objekte in Gruppen, die so genannten Cluster (engl. = Traube, Gruppe, Bündel) einzuteilen. Dabei sollen sich die Objekte in den Gruppen möglichst ähnlich sein (große intra-Cluster Ähnlichkeit), zugleich aber sollen die verschiedenen Cluster gut voneinander separiert sein (d.h. eine geringe inter-Cluster Ähnlichkeit aufweisen).

Die Cluster-Analyse findet in vielen Bereichen Anwendung, so dass hier nur exemplarisch einige wenige aufgezählt werden (vgl. Anderberg (1972, 5 f.) und Han und Kamber (2001, 336)):

- Marketing (Kundengruppen mit ähnlichen Interessen z.B. anhand des Einkaufsverhaltens ermitteln)

- Biologie (Taxonomien von Lebewesen erstellen, Gene mit ähnlichen Funktionen ermitteln)
- Geographie (Gebiete mit ähnlicher Bodennutzung anhand von Satellitenfotos identifizieren)
- Dokumente aus dem World Wide Web zur Informationsaufbereitung klassifizieren

2.2 Verwandte Gebiete und Abgrenzung zur Klassifikation

Die Cluster-Analyse gehört zu den multivariaten Analyseverfahren, da im Gegensatz zu den uni- oder bivariaten Verfahren nicht nur eine oder zwei Variablen betrachtet, sondern gleichzeitig die Beziehungen zwischen mehreren Variablen analysiert werden (vgl. Steinhausen und Langer 1977, 25). Zur Stellung der Cluster-Analyse innerhalb der multivariaten Analyseverfahren vgl. Ludwig (1994, 38 ff.).

Ein großer Unterschied besteht zur Klassifikation, dem so genannten „überwachten Lernen“: Hierbei werden Objekte oder Instanzen einer bereits definierten Klasse bzw. Gruppe zugeordnet. Im Gegensatz dazu sind bei der Cluster-Analyse die Klassen und deren Anzahl nicht a priori bekannt und werden erst durch das Verfahren selbst ermittelt. Bei der Klassifikation erfolgt die Zuordnung zu einer bestehenden Klasse z.B. im Rahmen der Diskriminanzanalyse, bei der die Elemente mit möglichst hoher Wahrscheinlichkeit einer bestimmten Klasse zugeordnet werden sollen (vgl. Steinhausen und Langer 1977, 12). Bei der Cluster-Analyse wird auf eine Vielzahl von Verfahren zur Ähnlichkeits- oder Distanzberechnung zurückgegriffen, um „natural groups“ (Anderberg 1972, 3) in den Ausgangsdaten zu ermitteln.

Insgesamt gesehen stellt die Cluster-Analyse ein Mittel zur explorativen Datenanalyse dar. Vor allem im Bereich des Data Minings kommt den clusteranalytischen Verfahren eine große Bedeutung zu, um eventuell vorhandene Strukturen in großen Datenmengen automatisch zu entdecken.

2.3 Ablauf einer Cluster-Analyse

Der Ablauf einer Cluster-Analyse wird von Steinhausen und Langer in mehrere Abschnitte untergliedert (vgl. Steinhausen und Langer 1977, 19 ff.). Nachfolgend werden diese Abschnitte grob charakterisiert, um einen Überblick über das allgemeine Vorgehen bei einer Cluster-Analyse zu erhalten. Eine ausführliche Beschreibung der einzelnen Abschnitte erfolgt in den weiteren Kapiteln dieser Arbeit, deren Reihenfolge sich an diesem Ablauf orientiert. Eine Cluster-Analyse beinhaltet folgende Schritte:

(1) Präzisierung der Untersuchungsfragestellung
(2) Auswahl der Elemente und Variablen
(3) Aufbereitung der Daten
(4) Festlegung einer angemessenen Ähnlichkeitsfunktion
(5) Bestimmung des geeigneten Algorithmus zur Gruppierung
(6) Technische Durchführung
(7) Analyse der Ergebnisse (Postanalyse)
(8) Interpretation der Ergebnisse

Die *Präzisierung der Untersuchungsfragestellung* soll den Anwender dazu bringen, den Einsatz von clusteranalytischen Verfahren hinsichtlich der generellen Eignung für einen bestimmten Problembereich zu überdenken. Bei der *Auswahl der Elemente und Variablen* soll der Anwender sicherstellen, dass diese für das Untersuchungsziel relevant und repräsentativ sind, um somit möglichen störenden Einflüssen vorzubeugen. Anschließend kann mit der *Aufbereitung der Daten* begonnen werden, bei der z.B. fehlende Werte ausgeschlossen werden oder eine Standardisierung der Daten durchgeführt wird. Die beschriebenen Schritte 1-3 werden von Jain et al. (vgl. Jain et al. 1999, 266 f.) unter *pattern representation* zusammengefasst. Dabei sollen durch *feature selection* Merkmale ausgewählt werden, die die Daten am geeignetsten charakterisieren. Mittels *feature extraction* sollen durch Umformung der Rohdaten neue (verdichtetere) Merkmale geschaffen werden, indem z.B. eine Faktor- oder Hauptkomponentenanalyse vorher durchgeführt wird. Die *Wahl eines Proximitätsmaßes* (Schritt 4) ist abhängig von der Domäne, innerhalb der die Clustering-Verfahren ihre Anwendung finden.

Wurden die Daten aufbereitet und ein geeignetes Proximitätsmaß ausgewählt, wird der eigentliche Gruppierungsvorgang durchgeführt, dem ein zuvor ausgewählter *(Fusionierungs-)Algorithmus* zu Grunde liegt (Schritt 5). Nach der *technischen Durchführung* erhält man eine Gruppierung, die durch die *Datenabstraktion* eine möglichst einfache und kompakte Beschreibung in Form von Cluster-Repräsentanten wie z.B. einem Centroid (Klassenschwerpunkt) liefern soll, um entweder von Menschen oder Computern weiterverarbeitet zu werden:

> „By data abstraction, we mean a simple and compact representation of the data. This simplicity helps the machine in efficient processing or a human in comprehending the structure in data easily.“ (Jain et al. 1999, 267)

Um Aussagen über die Güte der Ergebnisse treffen zu können, schließt sich eine *Analyse der Ergebnisse* an. Steinhausen und Langer verstehen darunter zunächst eine Beurteilung hinsichtlich

- „der Homogenität der gebildeten Cluster

- der Differenz der Clustermittelpunkte
- des Einflusses bestimmter Variablen und Element oder
- der Bedeutung der Startnäherung.“ (Steinhausen und Langer 1977, 21)

Im letzten Schritt findet die *Interpretation der Ergebnisse* statt. Jain et al. fassen Schritte 7 und 8 unter *„assessment of output“* zusammen, wobei ein für die jeweilige Domäne nützliches Gütemaß festgelegt werden muss, um die *Cluster Validity* der Lösung zu beurteilen (vgl. Kapitel 9).

3 Anwendungsbereich Patentrecherche

Der Anwendungsbereich, der dieser Arbeit zu Grunde liegt, wird in diesem Kapitel beschrieben, um ein Verständnis für die Besonderheiten und speziellen Anforderungen dieses Fachgebiets zu schaffen. Grundlagen des Patentwesens, wie z.B. die Voraussetzungen zur Patenterteilung und die Funktion von Patenten, werden aufgezeigt. Im weiteren Verlauf dieses Kapitels wird auf die Bedeutung von Patentrecherchen und die unterschiedlichen Motive dafür eingegangen. Zudem wird die für diese Zwecke vorhandene Infrastruktur vorgestellt, die z.B. in Form von Online-Datenbanken vorhanden ist.

3.1 Grundlagen des Patentwesens

In diesem Kapitel werden Grundlagen des nationalen Patentwesens der Bundesrepublik Deutschland vorgestellt, wie z.B. die Funktion von Patenten, die Voraussetzung zur Patenterteilung, der formale Aufbau einer Patentschrift und der Ablauf der Patenterteilung. Das Kapitel schließt mit einer Betrachtung der Sprache und des Stils von Patentschriften.

3.1.1 Funktion von Patenten

Ein Patent hat eine Doppelfunktion, bestehend aus einer Schutz- und Informationsfunktion: Die **Schutzfunktion** ist in § 9 des Patentgesetztes (PatG) formuliert („Das Patent hat die Wirkung, daß allein der Patentinhaber befugt ist, die patentierte Erfindung zu benutzen.“, Patentgesetz) und ermöglicht dem Patentinhaber ein zeitlich befristetes Monopolrecht zur Nutzung (maximal 20 Jahre). Es bietet ihm somit Schutz vor gewerblicher Nachahmung.

Der Staat schützt die gemachten Erfindungen vor direkter Nachahmung, jedoch „muss der Erfinder, sozusagen als Gegenleistung, seine Erfindung der Allgemeinheit preisgeben und erhöht somit den Stand der Technik.“ (Wurzer 2003, 49) Das stellt die **Informationsfunktion** von Patenten dar. Durch das öffentlich verfügbare Wissen sollen Innovationen und der technische Fortschritt angeregt werden. Diese Grundidee spiegelt sich in der Etymologie des Wortes „Patent“ wider: Für „patere“ wird als Übersetzung „offen legen“ und nicht „schützen“ angegeben (vgl. Wurzer 2003, 49).

3.1.2 Voraussetzungen für ein Patent

Patente können für viele Bereiche erteilt werden, so beispielsweise für:

- ❏ technische Gegenstände und Verfahren (Maschinen, Vorrichtungen, Geräte und deren Teile)
- ❏ chemische Erzeugnisse
- ❏ Arzneimittel
- ❏ Verfahren zum Herstellen von Erzeugnissen, Arbeits- und Anwendungsverfahren
- ❏ mikrobiologische Verfahren und deren Anwendung.

Daneben gibt es Bereiche, für die keine Patente erteilt werden dürfen (so z.B. die in § 1 Abs. 2 und § 2 PatG genannten Bereiche). Darunter fallen z.B. (Göbel, o.J.):

- ❏ ästhetische Formschöpfungen (Design)
- ❏ Regeln für Spiele und reine EDV-Programme (Software)
- ❏ Entdeckungen sowie wissenschaftliche Theorien und mathematische Methoden
- ❏ Pflanzensorten oder Tierarten
- ❏ Verfahren zur chirurgischen oder therapeutischen Behandlung des menschlichen oder tierischen Körpers und Diagnostizierverfahren

Laut § 1 des Patentgesetzes muss eine Erfindung drei Voraussetzungen erfüllen, um patentfähig zu sein:

1. Es muss sich um eine (weltweite) Neuheit handeln (vgl. § 3 PatG).
2. Dem zu patentierenden Gegenstand muss eine erfinderische Tätigkeit zu Grunde liegen (vgl. § 4 PatG).
3. Die Erfindung muss eine (denkbare) gewerbliche Anwendung ermöglichen (vgl. § 5 PatG).

Eine Neuheit liegt dann vor, wenn ein Gegenstand nicht zum Stand der Technik (d.h. sämtliches derzeit verfügbares technisches Wissen) gehört und zuvor nichts darüber veröffentlicht wurde (z.B. als Beschreibung in einem Vortrag oder einer Publikation) (vgl. Göbel, o.J.). Man spricht von einer erfinderischen Tätigkeit, wenn die „Erfindung keine einem [fiktiven] Fachmann naheliegende Weiterentwicklung des Standes der Technik darstellt“ (vgl. Wurzer 2003, 54) und somit die nötige *Erfindungshöhe* für ein Patent aufweist.

3.1.3 Aufbau einer Patentschrift

Schramm (2004, 89) unterscheidet zwischen einem Patentdokument und einer Patentschrift. *Patentschriften* sind Dokumente, die von den (inter-)nationalen Paten-

tämtern veröffentlicht werden und die die von Gesetzes wegen erforderlichen Angaben zur Anmeldung eines Patents enthalten. *Patentdokumente* stellen über die Patentschriften hinausgehende Informationen bereit, wie z.B. Sekundärliteratur in Form von Patentreferaten (Abstracts) und Informationen über den Verfahrensstand von Patenten. Die erste Seite einer Patentschrift (Deckblatt und Teile der Beschreibung sowie der Patentansprüche, vgl. Abbildung 3.1) enthält Informationen zu:

- ❏ Titel
- ❏ Zusammenfassung
- ❏ Namen (des Erfinders, des Anmelders, des Patentanwalts)
- ❏ Daten (Anmeldedatum, Publikationsdaten z.B. Tag der Offenlegung)
- ❏ Nummern (z.B. Publikationsnummer)
- ❏ Zeichnung des Patentgegenstandes (nicht verpflichtend)

Die Nummern in den Kreisen auf dem Deckblatt identifizieren (zusätzlich zur Benennung) die bibliographischen Daten, wobei die verwendeten Codes international normiert sind (INID-Code). Das soll Suchenden helfen, die für sie relevanten Angaben zu ermitteln, falls sie die jeweilige Sprache oder Gesetzesgrundlage (auf deren Basis eine Angabe erforderlich ist) nicht kennen.

Im Hauptteil einer Patentschrift (siehe Abbildung 3.2) werden Hintergrundinformationen, die erfinderischen Einzelheiten und die Ansprüche (Haupt- und Nebenansprüche) dargelegt. Von besonderer Bedeutung sind dabei die Ansprüche, da in ihnen das Neue und Einzigartige der Erfindung aufgeführt wird, was zur Charakterisierung und Abgrenzung der zu patentierenden Gegenstände herangezogen wird (vgl. Thomä und Tribiahn 2002, 8f.).

(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) **Patentschrift**

(10) **DE 195 27 841 C 1**

(51) Int. Cl.[6]:
B 65 B 3/00
B 65 B 3/04
B 65 B 43/54
B 67 C 3/02
B 67 C 3/24

(21) Aktenzeichen: 195 27 841.0-27
(22) Anmeldetag: 29. 7. 95
(43) Offenlegungstag: –
(46) Veröffentlichungstag der Patenterteilung: 14. 8. 96

DE 195 27 841 C1

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

(73) Patentinhaber:
Hamba-Maschinenfabrik Hans A. Müller GmbH & Co KG, 42275 Wuppertal, DE

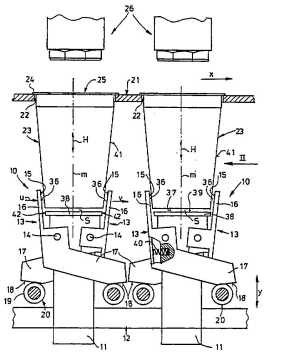
(74) Vertreter:
Patentanwälte Ostriga, Sonnet & Wirths, 42275 Wuppertal

(72) Erfinder:
Biehl, Fritz, 66539 Neunkirchen, DE

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:
DE 25 09 611 A1

(54) Becherfüllwerk für Nahrungs- und Genußmittel, insbesondere für dünnflüssige bis pasteuse Molkerei- und Fettprodukte

(57) Bei einem Becherfüllwerk für Nahrungs- und Genußmittel, insbesondere für dünnflüssige bis pasteuse Molkerei- und Fettprodukte, sind Becher (23) über eine Becherhebevorrichtung mit Hebetellern (10) aus Becheraufnahmen (22) heraus in ihre Arbeitsposition anhebbar und anschließend wieder in die Becheraufnahmen (22) absenkbar. Eine feste Becherhalterung für die angehobene Becherposition wird dadurch erzielt, daß jedem Hebeteller (10) zwei einander gegenüberliegende, beiderseits der vertikalen Bewegungsachse (H) des Hebetellers (10) angeordnete, gemeinsam mit dem Hebeteller auf- und abbewegliche Klemmbacken (13) zugeordnet sind, welche mit ihren der Becheraußenmantelfläche (41) zugewandten Klemmflächen (15) bewegungseinheitlich in Schließrichtung (u) auf den Becher (23) zustellbar und in Richtung (v) Offenstellung von diesem (23) wegbewegbar sind und welche in Offenstellung eine sich in Förderrichtung erstreckende Becher-Durchtrittsgasse (36) begrenzen.



DE 195 27 841 C1

BUNDESDRUCKEREI 06. 96 602 133/394

Abbildung 3.1: Deckblatt eines Patents

41 des Bechers 23 an.

Fig. 4 zeigt, daß die Anordnung 10 ihre höchste Vertikalposition entlang der Bewegungsachse H erreicht hat, in welcher der Doppelfüllstutzen 26 bereits sehr tief in den Innenraum des Bechers 23 eingetaucht ist. Während des Füllvorganges dreht sich der Doppelfüllstutzen 26 um die Drehachse A, wobei sich der Hebeteller 10 mitsamt dem Becher 23 wieder nach unten senkt. Dabei wird eine aus zwei verschiedenfarbigen Produkten gebildete wendel- bzw. schraubenlinienartig eingeschichtete Füllung erzeugt.

Bei einem weiteren Absenken der Anordnung 10 wird schließlich die Vertikalposition gemäß **Fig.** 3 erreicht, gemäß welcher sich der Becherflansch 24 oben auf dem Zellenbrett 21 abstützt. Bei einer weiteren Absenkbewegung rutschen schließlich die Klemmflächen 15 an der Außenmantelfläche 41 des Bechers 23 um ein geringes Höhenmaß nach unten, bis die Steuerflächen 18 die ortsfesten Anschlagrollen 19 berühren, wonach bei einer weiteren Abwärtsbewegung der Hebeteller 10 sich die Klemmbacken 13 in Richtung v öffnen, so daß wiederum der Zustand gemäß **Fig.** 1 erreicht ist.

Fig. 5 zeigt daß der Tellerflansch 38 kreisförmiger Grundform an diametral gegenüberliegenden Umfangsbereichen 42 keisabschnittartig abgearbeitet (abgeflacht) ist, so daß die Klemmflächen 15 die Außenmantelfläche 41 des Bechers 23 besser ergreifen können.

Im Zusammenhang der Zeichnungen wird auch deutlich, daß jeweils beide Betätigungsarme 17 einer Anordnung 10, in Ansicht der Anordnung 10 betrachtet, einander kreuzen, womit eine kompakte Bauform erzielt wird. **Fig.** 4 zeigt in diesem Zusammenhang, daß die geschlossenen Klemmbacken 13 mit ihren Klemmarmen 16 innerhalb der Axialprojektion des größten Becherquerschnittes (Becherflansch 24 ausgenommen) angeordnet sind, so daß die Klemmarme 16 von unten her durch die Becheraufnahmen 22 hindurchtauchen können.

Fig. 2 zeigt den Hubbalken 12 und eine Traverse 43 zur Befestigung des Hebeteller-Schaftes 11 in durchgezogener Stellung (untere Position) und in strichpunktierter Darstellung (höchste Vertikalposition gemäß **Fig.** 4).

Aus **Fig.** 2 ist auch zu ersehen, daß die raumfesten Anschläge 20 an einer Konsole 44 befestigt sind und daß die Konsole 44 in nicht näher gezeichneter Weise über Langlochanordnungen 45 höhenverstell- und arretierbar ist. Auf diese Art kann die Klemmweite zwischen zwei einander gegenüberliegenden Klemmflächen 15 verändert bzw. justiert werden.

Patentansprüche

1. Becherfüllwerk für Nahrungs- und Genußmittel, insbesondere für dünnflüssige bis pasteuse Molkerei- und Fettprodukte od. dgl., mit einem umlaufend geführten Fördermittel, welches quer (bei x) zur Förderrichtung in Reihen angeordnete Becheraufnahmen (22) für die Becher (23) trägt, die außerdem parallel zur Förderrichtung angeordnete Bahnen bilden und mit ihnen ausgerichtete Arbeitsstationen, insbesondere eine Becher-Zuführstation, eine Produkt-Doseurstation (bei 27), eine Deckel-Auflegestation, eine Deckel-Verschließstation, wie z. B. Siegel- bzw. Schweißstation, sowie eine Becher-Entnahmestation in zeitlich gleichen Vorschubtakten aufeinanderfolgend durchlaufen, wobei jeder mit einer Arbeitsstation, wie z. B. mit einem Produktdoseur (27), ausgerichtete Becher (23) jeweils durch einen Hebeteller (10) einer raumfest angeordneten Becherhebevorrichtung in seine Arbeitsposition, wie z. B. in seine Füllposition, anhebbar und anschließend wieder in die Becheraufnahme (22) absenkbar ist, **dadurch gekennzeichnet,** daß jedem Hebeteller (10) zwei einander gegenüberliegende, beiderseits der vertikalen Bewegungsachse (H) des Hebetellers (10) angeordnete, gemeinsam mit dem Hebeteller (10) auf- und abbewegliche Klemmbacken (13) zugeordnet sind, welche mit ihren der Becheraußenmantelfläche (41) zugewandten Klemmflächen (15) bewegungseinheitlich in Schließrichtung (u) auf den Becher (23) zustellbar und in Richtung (v) Offenstellung von diesem wegbewegbar sind und welche in Offenstellung eine sich in Förderrichtung erstreckende Becher-Durchtrittsgasse (36) begrenzen.

2. Becherfüllwerk nach Anspruch 1, dadurch gekennzeichnet, daß die Klemmflächen (15) eben sind.

3. Becherfüllwerk nach Anspruch 1 oder nach Anspruch 2, dadurch gekennzeichnet, daß die Klemmflächen (15) der Neigung der Mantellinie der Becheraußenmantelfläche (41) angepaßt sind.

4. Becherfüllwerk nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß die Klemmbacken (13) in Zustellrichtung (u) rückstellbelastet sind.

5. Becherfüllwerk nach Anspruch 4, dadurch gekennzeichnet, daß die Klemmbacken (13) in Zustellrichtung (u) rückstellfederbelastet sind.

6. Becherfüllwerk nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß die Klemmbacken (13) am Hebeteller (10) bewegungsgelagert sind.

7. Becherfüllwerk nach Anspruch 6, dadurch gekennzeichnet, daß die Klemmbacken (13) am Hebeteller (10) schwenkgelagert sind.

8. Becherfüllwerk nach Anspruch 7, dadurch gekennzeichnet, daß jede Klemmbacke (13) Bestandteil eines am Hebeteller (10) schwenkbar gelagerten doppelarmigen Hebels ist, dessen die Klemmfläche (15) tragender einer Arm einen Klemmarm (16) und dessen anderer Arm den Betätigungsarm (17) bildet.

9. Becherfüllwerk nach Anspruch 8, dadurch gekennzeichnet, daß der jeweilige Betätigungsarm (17) in unterster Stellung des Hebetellers (10), also in Offenstellung der Klemmbacken (13), gegen einen raumfesten Anschlag (20) drückt.

10. Becherfüllwerk nach Anspruch 9, dadurch gekennzeichnet, daß der raumfeste Anschlag (20) als Rolle (19) ausgebildet ist, welche sich bei der Vertikalbewegung (bei y) des Hebetellers (10) am Betätigungsarm (17) abwälzt.

11. Becherfüllwerk nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, daß Klemmarm (16) und Betätigungsarm (17) miteinander einen stumpfen Winkel einschließen.

12. Becherfüllwerk nach Anspruch 11, dadurch gekennzeichnet, daß beiderseits der Bewegungsachse (H) des Hebetellers (10) je ein Betätigungsarm (17) angeordnet ist, daß sich beide Betätigungsarme (17), in Ansicht des Hebetellers (10) betrachtet, einander kreuzen, und daß die Federrückstellkraft jeweils von einer Druckfeder (40) gebildet ist, deren eines Ende im Hebeteller (10) gelagert ist und deren anderes Ende den Betätigungsarm (17) nach außen drückt.

13. Becherfüllwerk nach einem der vorangehenden

Abbildung 3.2: Ausschnitt aus dem Beschreibungsteil und dem Hauptanspruch einer Patentschrift

3.1.4 Klassifikation von Patentschriften

Um Patente besser auffindbar zu machen, wird jeder Patentschrift inhaltlich, auf Basis des Hauptanspruchs, eine Klasse zugewiesen. In der Bundesrepublik Deutschland wird, wie international weitestgehend üblich, als Systematik die IPC (International Patent Classification)[1] eingesetzt. Sie liegt unter der Verantwortung der WIPO (World Intellectual Property Organization) und wird alle fünf Jahre in revidierter Fassung herausgegeben. Zurzeit gültig ist die IPC in der Version 7 (2000-2004), die über 60.000 Teilgebiete umfasst. Eine Weiterentwicklung der IPC ist nötig, weil z.B. neue Erfindungen gemacht werden, die sich nicht in existierende Klassen einordnen lassen (vgl. Wittmann 1992, 88). Bereits klassifizierte Dokumente werden nicht umklassifiziert, daher muss man (z.B. bei Recherchen) die entsprechend gültige IPC-Version zum Zeitpunkt der Patenterteilung berücksichtigen (vgl. Thomä und Tribiahn 2002, 11). Die IPC ist ein hierarchisch aufgebautes Klassifikationssystem, das auf der höchsten Hierarchiestufe acht Sektionen enthält (siehe Tabelle 3.1).

A	Täglicher Bedarf
B	Arbeitsverfahren, Transportieren
C	Chemie, Hüttenwesen
D	Textilien, Papier
E	Bauwesen, Erdbohren, Bergbau
F	Maschinenbau, Beleuchtung, Heizen, Waffen, Sprengen
G	Physik
H	Elektrotechnik

Tabelle 3.1: Sektionen der IPC

Von den Sektionen erfolgt eine weitergehend detailliertere Einteilung über Klassen, Unterklassen, Hauptgruppen und Untergruppen (vgl. Tabelle 3.2). Bei der Einteilung eines Erfindungsgegenstandes wird nach Funktion und Anwendung unterschieden, wobei gilt:

> „Er wird in eine ‚allgemeine Klasse' eingeordnet, wenn er in verschiedenen Anwendungsgebieten einsetzbar ist. Ist ein Erfindungsgegenstand dagegen besonders für eine bestimmte Anwendung ausgebildet, so wird er in eine ‚Spezialklasse' eingeordnet." (Wittmann 1992, 88)

Kann der Inhalt eines Patents nicht durch eine Klasse vollständig ausgedrückt werden, können zu der Hauptklassifikation (MainIPC) weitere Klassen angegeben werden (Nebenklassen).

[1] Zur Entstehungsgeschichte der IPC wird auf Wittmann (1992, 81 f.) verwiesen.

Hierarchiestufe	Symbol	Beschreibung
Sektion	G	Physik
Klasse	G06	Datenverarbeitung; Rechnen; Zählen
Unterklasse	G06F	Elektrische digitale Datenverarbeitung
Hauptgruppe	G06F017	Digitale Rechen- oder Datenverarbeitungsanlagen oder -verfahren
Untergruppe	G06F017/30	Wiederauffinden von Informationen; Struktur der Datenbasis dafür

Tabelle 3.2: Beispiel für den hierarchischen Aufbau der IPC

3.1.5 Stationen auf dem Weg zum Patent

Nachdem die Patentanmeldung beim Deutschen Patent- und Markenamt (DPMA mit Sitz in München) eingegangen ist, wird eine Anmeldenummer (= Anmeldeaktenzeichen) vergeben. Ist die Anmeldung formal korrekt und vollständig *(Offensichtlichkeitsprüfung)*, wird dem Antrag anhand der internationalen Patentklassifikation eine Klasse zugewiesen, die den technischen Bereich der Erfindung charakterisiert. Achtzehn Monate nach Patentanmeldung wird der Inhalt in der sog. Offenlegungsschrift (= ungeprüfte Anmeldeschrift) publiziert und erhält eine Patentnummer.

Eine Prüfung der Patentanmeldung findet (in Deutschland) nur auf Antrag statt. Wird innerhalb von sieben Jahren nach der Anmeldung kein Antrag auf Prüfung gestellt, gilt die Anmeldung als zurückgenommen. Sind bei der beantragten *Sachprüfung* alle inhaltlichen Voraussetzungen (vgl. Kapitel 3.1.2) für ein Patent erfüllt, wird das Patent erteilt und zusammen mit seiner Patentnummer als Publikationsnummer im deutschen Patentblatt veröffentlicht. Den Status eines Patents (Offenlegungschrift oder bereits erteiltes Patent) kann man z.B. in der Patentdatenbank PATDPA (vgl. Kapitel 3.2.4) abfragen (vgl. Thomä und Tribiahn 2002, 6ff.).

3.1.6 Sprachliche und stilistische Besonderheiten von Patentschriften

Die Sprache, die zur Beschreibung innerhalb von Patentschriften eingesetzt wird, weicht häufig in Stil und Vokabular von der Fachsprache ab, wie sie z.B. in wissenschaftlichen Publikationen im jeweiligen Fachgebiet vorherrscht. Zwei Erklärungsmöglichkeiten bieten sich an, warum die Patentanmelder häufig auf eine hohe Abstraktionsebene mit sehr allgemeinen Beschreibungen für ihren Gegenstand ausweichen (vgl. Krause 1987, 223). So wird z.B. aus einer „Mausefalle“ ein „Gerät zum Festsetzen kleiner Nagetiere“ (Wurzer 2003, 194). Durch diese möglichst allgemeine Art der Beschreibung:

1. soll es Konkurrenten nicht leicht gemacht werden, durch Patentrecherche Wettbewerbsanalyse zu betreiben und somit die Entwicklungstätigkeiten von Wettbewerbern nachzuvollziehen (vgl. Wurzer 2003, 193 f.).

2. soll ein Patent einen möglichst großen Schutzumfang aufweisen, so dass ein Patentinhaber z.B. unliebsame Konkurrenzprodukte auf Grund von Ähnlichkeiten mit einem eigenen Patent im Idealfall vom Markt drängen kann. Weiterhin soll vermieden werden, dass konkurrierende Firmen durch kleine Detailveränderungen neue Patente anmelden können (vgl. Krause 1987, 223).

Ein Hilfsmittel, um dieser sprachlichen Verschleierung entgegen zu wirken, ist die Patentklassifikation anhand der IPC oder eines anderen Klassifikationssystems. Bei der Suche innerhalb einer bestimmten Klasse können ähnliche Patente ermittelt werden, die bei einer Stichwortsuche nicht entdeckt worden wären. In der Datenbank WPINDEX (siehe Kapitel 3.2.3) wird zusätzlich ein anderer Weg gegangen: Dabei bilden nicht die Originaltitel und -abstracts die alleinige Textbasis für die Recherche, sondern die Patentschriften werden von Fachleuten gelesen und daraufhin werden neue Titel- und Abstract-Informationen erstellt. Die dabei verwendete Terminologie gleicht der im Fachbereich gängigen (vgl. Wurzer 2003, 193).

3.2 Patentrecherche: Gründe und Infrastruktur

Die Motive zur Patentrecherche und die dafür zur Verfügung stehende Infrastruktur sind Gegenstand dieses Kapitels. Zunächst wird die wirtschaftliche Bedeutung von Patenten charakterisiert, um anschließend die Einrichtungen und Zugangsmöglichkeiten zu Patentinformationen zu beschreiben.

3.2.1 Die wirtschaftliche Bedeutung von Patenten

Patentschriften bieten eine große Aktualität und beinhalten aufgrund der detaillierteren Beschreibungen mehr Informationen, als es z.B. in Fachzeitschriften der Fall ist. Zudem sind ca. 90 % der Patente frei verwertbar, da deren Patentschutz erloschen ist (Göbel, o.J.). Damit stellt die Patentliteratur eine der wichtigsten technischen Informationsquellen dar, da

> „[...] etwa 85 bis 90 Prozent des technischen Wissens in der Patentliteratur publiziert [wird]. Dabei werden nur etwa 5 bis 10 Prozent des in der Patentliteratur veröffentlichten Wissens in der sonstigen Literatur wiedergegeben, und das erfolgt oft erst bis zu fünf Jahre nach der Anmeldung des entsprechenden Patents.“ (Wurzer 2003, 27)

Durch die gezielte Nutzung von Patentinformationen eröffnet sich für ein Unternehmen eine Vielzahl interessanter Handlungsfelder, die für den unternehmerischen Erfolg von großer Bedeutung sein können. Einige Möglichkeiten werden im Folgenden exemplarisch vorgestellt.

Kommerzielle Verwertung durch Lizenzierung und Kooperation

Der Inhaber eines Patents genießt Schutz vor gewerblicher Nachahmung seiner Erfindung (vgl. Kapitel 3.1.1). Er kann entweder selbst die Idee in ein marktreifes Produkt umsetzen oder durch Lizenzvergabe einem Vertragspartner Nutzungsrechte für die Erfindung gewähren (vgl. Wurzer 2003, 56). Die dafür fälligen Lizenzgebühren stellen eine wichtige Einnahmequelle für den Schutzrechtsinhaber dar.

Neben der Lizenzierung als Verwertungsstrategie bieten sich auch Kooperation und die Einräumung gegenseitiger Nutzungsrechte an patentierter Technologie an, um wirtschaftlichen Nutzen zu erzielen (Patente als Mittel der strategischen Unternehmensplanung, siehe Beispiel in Wurzer 2003: 28 ff.).

Die Wichtigkeit des Patentbesitzes (vor allem im zunehmenden internationalen Wettbewerb) lässt sich anhand der Patentstatistik des Europäischen Patentamtes verdeutlichen. Im Jahre 2001 stammten fast 30 % aller anmeldenden Unternehmen aus den USA, deutsche Unternehmen machten knapp 20 % aus, und der Anteil der japanischen Unternehmen belief sich auf ca. 18 %. Das Dilemma, in dem sich Unternehmen befinden, wird von Prof. Erich Hauser, ehem. Präsident des DPMA, kurz folgendermaßen skizziert: „Wer nicht erfindet, verschwindet. Wer nicht patentiert, verliert. Und wer sich nicht informiert, der stirbt.“ (zitiert nach Wurzer 2003, 28)

Aufspüren von technologischen Trends

Mittels Patentinformationen können frühzeitig[2] technologische Trends in abgegrenzten Technologiefeldern ermittelt werden, was es einem Unternehmen ermöglicht, gezielt darauf zu reagieren, bevor diese Trends durch Veröffentlichungen zu Allgemeingut werden (vgl. Wurzer 2003, 64).

Wettbewerberanalyse

Patentinformationen können als Mittel zur Analyse von Wettbewerberaktivitäten eingesetzt werden. Vor allem im Vergleich mit den eigenen Forschungs- und Entwicklungsaktivitäten (F&E-Aktivitäten), der Innovationskraft und der Positionierung im Wettbewerb eines Unternehmens können anhand von Patentanalysen die Technolgieführer identifiziert werden, um „deren F&E-Politik in wichtigen Schlüssel- und Zukunftstechnologien zu studieren“ (Wurzer 2003, 67). (Siehe hierzu auch Schramm 2004, 101 ff.)

Planung von Forschungs- und Entwicklungsaktivitäten

Auf der Basis von Patentinformationen lässt sich besser abschätzen, ob ein Unternehmen selbst Ressourcen in die Entwicklung von Technologie stecken möchte, oder

[2] „Erfindungen werden erfahrungsgemäß vier bis sieben Jahre vor Beginn ihrer wirtschaftlichen Nutzung zum Patent angemeldet.“ (Wittmann 1992, 175).

ob die benötigte Technologie eventuell günstiger durch Lizenzerwerb beschafft werden kann. Außerdem lassen sich wichtige Schlüsselpersonen und deren Forschungsschwerpunkte identifizieren (vgl. Wurzer 2003, 67 f.).

Gerade im Zusammenhang mit Forschungs- und Entwicklungstätigkeiten ist eine Patentrechereche unabdingbar, wobei Schramm hierbei zwischen drei typischen Recherchearten unterscheidet: die Weltstands-, Neuheits- und Verletzungsrecherche, die sich im Grad der Retrospektivität und Recherchevollständigkeit unterscheiden (siehe Tabelle 3.3, wobei PCT = Patent Cooperation Treatey – Vertrag über die internationale Zusammenarbeit auf dem Gebiet des Patentwesens, vgl. hierzu Wittmann (1992, 24 ff.)). Vor Beginn einer F&E-Arbeit wird bei der *Weltstandsrecherche* ermittelt, ob es bereits vorhandene Patente gibt, um so unnötige und kostspielige Doppelforschung zu vermeiden. Bei der *Neuheitsrecherche* wird auch Sekundärliteratur (Nichtpatentliteratur) berücksichtigt, um zu entscheiden, ob eine Erfindung generell patentierbar ist. Bei der *Verletzungsrecherche* wird ermittelt, ob durch Benutzung, Produktion und Vertrieb technischer Lösungen fremde Patentrechte beeinträchtigt werden (vgl. Schramm 2004, 96).

Rechercheart	**Retrospektivität**	**Länderspektrum**	**Rechercherchevollständigkeit**
Weltstandsrecherche	5–10 Jahre	PCT-Minimaldokumentation	nicht notwendig
Neuheitsrecherche	bis 1920	PCT-Minimaldokumentation	unbedingt notwendig
Verletzungsrecherche	15–25 Jahre	Konkurrenz-/ Exportländer	unbedingt notwendig

Tabelle 3.3: Arten der Patentrecherche (Schramm 2004, 97)
(PCT-Minimaldokumentation = Länder in empfohlener Suchreihenfolge: DE, EP (Europäisches Patentamt), WO (=WIPO), US, JP, RU, GB, FR

3.2.2 Das FIZ-Karlsruhe und seine Rolle in der Bereitstellung von Patentinformationen

Das Fachinformationszentrum Karlsruhe (FIZ-Karlsruhe) erfüllt zahlreiche an Dienstleistungsaufgaben im Rahmen der Informationsversorgung[3]. Beispielsweise werden auf den Rechnern des FIZ-Karlsruhe verschiedene Datenbanken technisch administriert und für Online-Recherchen ständig verfügbar gehalten, wie z.B. die Patentdatenbanken des DPMA. Diese Funktion wird als Host bezeichnet.

Als Host ist das FIZ-Karlsruhe neben dem Aufrechterhalten des laufenden Datenbankbetriebs auch verantwortlich für die Nutzerverwaltung, z.B. durch Erteilung von

[3]Die Patentinformation ist nur eine Teilaufgabe des FIZ-Karlsruhe. Generell hat das FIZ-Karlsruhe zum Ziel, wissenschaftlich-technische Informationsdienste für Forschung und Lehre, Wissenschaft und Wirtschaft, Technik und Verwaltung bereitzustellen. Zu einer Beschreibung der Aufgabenschwerpunkte und Tätigkeiten vgl. FIZ-Karlsruhe (2000).

Zugangsberechtigungen zu den verschiedenen Datenbanken, für die Bereitstellung des Zugangs zu weiteren Datenbanken im Rahmen von Kooperationen beispielsweise über den Verbund STN International (Scientific & Technical Information Network) und für die Abrechnung von kostenpflichtigen Diensten und Leistungen.

Über den Verbund STN International kann auf 220 Datenbanken zugegriffen werden, deren Umfang enorm ist: „Es werden mehr als 370 Millionen Zitate und chemische Strukturen, 35 Millionen Patentdokumente, 15 Millionen Patentfamilien und 59,3 Millionen Rechtsstandstaten nachgewiesen." (Wurzer 2003, 176)

3.2.3 Online Patentdatenbanken

Online-Datenbanken werden über Hosts (wie z.B. das FIZ-Karlsruhe, siehe vorheriges Kapitel) angeboten. Die Inhalte der Datenbanken werden von verschiedenen Produzenten geliefert, so ist z.B. das Deutsche Patent- und Markenamt Produzent für die Datenbank PATDPA (Patentdaten des Deutschen Patent- und Markenamts, vgl. hierzu Kapitel 3.2.4).

Neben den nationalen oder internationalen Patentämtern (wie dem Europäischen Patentamt, EPA bzw. EP) gibt es weitere kommerziell orientierte Produzenten von Datenbankinhalten. Als Beispiel sei hier das Unternehmen Thomson-Wila-Derwent genannt, welches zwar ebenfalls auf die Originaldaten der Patentämter zugreift, diese aber durch Hinzufügen von Mehrwerten veredelt. So werden „Sekundärinformationen wie insbesondere strukturiert aufgebaute Abstracts erstellt..." (Wittmann 1992, 136) oder Patentinformationen aus anderen Sprachen (z.B. aus dem Japanischen oder Chinesischen) ins Englische übersetzt, um mit Englisch als Lingua-Franca einen Zugriff auf diese Inhalte zu ermöglichen (vgl. Wittmann 1992, 145). Als Beispiel kann hierzu die Datenbank WPINDEX (Derwent World Patents Index) von Thomson Scientific (London) betrachtet werden.

> „Sie enthalten bibliographische Daten und Abstracts von Patentdokumenten, die von 29 nationalen Patentämtern sowie vom Europäischen Patentamt und der WIPO herausgegeben wurden. Die Abstracts werden nach vorgegebenen Regeln auf der Grundlage des vollständigen Patentdokuments erstellt." (Wittmann 1992, 149 f.)

Einer strikten Zuordnung zu bibliographischen oder Faktenbanken entziehen sich Patentdatenbanken auf Grund ihrer Inhalte. Manche Patentdatenbanken liefern bibliographische Verweise auf Volltexte von Patenten (z.B. PATDPA), andere wiederum beinhalten die Volltexte selbst (z.B. PATDPAFULL). Außerdem enthalten Patentdatenbanken zugleich Informationen, die als Fakten anzusehen sind, bspw. die Daten zum Stand des Verfahrens von Patentanmeldungen (vgl. Wittmann 1992, 142).

Die über den Verbund STN International verfügbaren Datenbanken können über eine einheitliche Kommandosprache namens „Messenger“ abgefragt werden, was für einen Benutzer viele Vorteile bringt. Mit nur einer Kommandosprache kann er in mehreren Datenbanken recherchieren, ohne jeweils eine separate Befehlssyntax erlernen zu müssen.

Die Nutzer von Patentdaten stammen aus Wissenschaft, Verwaltung und Industrie (vgl. Kapitel 3.2.1). Daneben sind es die Patentämter selbst, die im Rahmen von Sachprüfungsverfahren relevante Patentliteratur ermitteln müssen, um den zu prüfenden Gegenstand mit dem aktuellen Stand der Technik zu vergleichen.

3.2.4 Die Datenbank PATDPA

Bei der Datenbank PATDPA des DPMA handelt es sich um eine Fortschreibungsdatenbank. Bei jeder neu zu berücksichtigenden Veröffentlichung wird eine eigene Dokumentationseinheit (d.h. Patentdokument mit eigener Systemnumber) angelegt. Änderungen des Verfahrensstandes eines Dokuments wie z.B. Offenlegungsschrift, Prüfungsantrag gestellt, Patent erteilt u.a, werden der Dokumentationseinheit hinzugefügt. Dieses Prinzip wird als *dynamische Fortschreibung* bezeichnet. Dabei gilt: „Eine Dokumentationseinheit entspricht einem bestimmten Verfahren von einem der Ämter der drei Patentorganisationen (DPMA, EPA, WIPO), unabhängig von der Anzahl der Publikationen durch die betreffende Organisation.“ (Thomä und Tribiahn 2002, 56) Wird ein Patent in mehreren Ländern angemeldet, so spricht man von einer Patentfamilie.

Für jede Anmeldung entsteht folglich ein neuer Datensatz, dessen Status an den jeweiligen Bearbeitungsstand angeglichen wird. Die Datenbank enthält bibliographische Informationen (vgl. Kapitel 3.1.3) zu allen im „Patentblatt“ veröffentlichten deutschen Offenlegungs-, Auslege-, Patent- und Gebrauchsmusterschriften sowie den Patentveröffentlichungen des Europäischen Patentamtes und der Weltorganisation für Geistiges Eigentum (WIPO) mit Bestimmung der Bundesrepublik Deutschland als Vertragsstaat. Die Erfassung beginnt im Jahre 1968. Bislang sind über 7,41 Millionen Zitate und über 490.000 Patentzeichnungen hinterlegt (Stand: Dezember 2002) (vgl. Wurzer 2003, 180 f.). Eine Beschreibung der Datenbank und deren recherchierbare Felder ist bei den STN-Datenbanken über ein so genanntes „Database Summary Sheet“ einsehbar. Für die Datenbank PATDPA ist dieses Database Summary Sheet unter `http://www.cas.org/ONLINE/DBSS/patdpass.html` abrufbar.

Bis zum Jahre 1998 wurden die Patentdokumente zur Inhaltserschließung zusätzlich im Feld „PST“ der Datenbank mit Termen versehen (vgl. TU Ilmenau, 6). Die Terme wurden mit der Software PASSAT (erstellt von Siemens) auf Basis der Felder Titel und Abstract der Originaldokumente ermittelt (vgl. Wittmann 1992, 147).

Dabei wurden Wortformen auf ihre Grundform reduziert, Komposita in sinntragende Bestandteile zerlegt und einem Wort semantisch ähnliche Grundformen zugeordnet (z.B. „durch Kondensierung“ - Kondensierung, Kondensieren, Kondensation) (vgl. Bauer und Schneider 1990, 35). Dadurch sollten Benutzer beim Retrieval unterstützt werden, um relevante Ergebnisse ohne Kenntnis der exakten Vollformen im Ausgangstext zu erhalten.

3.3 Zusammenfassung

Patente weisen zugleich eine Schutz- und eine Informationsfunktion auf. Sie werden regelmäßig von den Patentämtern publiziert und stellen einen großen und wichtigen Teil der technischen Fachliteratur dar, der den aktuellen Stand der Technik widerspiegelt. Zur Recherche stehen Online-Patentdatenbanken zur Verfügung, die entweder kostenlos bei den Patentämtern oder kostenpflichtig, dafür aber mit einem Mehrwert ausgestattet, bei speziellen Hosts, wie z.B. dem FIZ-Karlsruhe, angesiedelt sind.

Durch die zunehmende und vielfältige wirtschaftliche Nutzung von Patentinformationen erfahren Werkzeuge zur Analyse von Patenten eine immer größere Bedeutung. Im folgenden Kapitel wird ein solches Werkzeug, das Clustern von Dokumenten, vorgestellt.

4 Clustering im IR und im Anwendungsbereich Patentrecherche

Die Darstellung der Einsatzmöglichkeiten von Clustering-Tools ist Gegenstand dieses Kapitels. Es wird ein Überblick über die verschiedenen Ansätze und Verwendungsmöglichkeiten gegeben, wobei die Darstellung sich zunächst allgemein im Bereich des Information Retrieval (IR) orientiert und anschließend auf das Anwendungsgebiet der Patentrecherche und Patentanalyse ausgedehnt wird. Zudem werden mit dem Themenbereich verbundene Probleme beschrieben, wie das automatische Generieren von Cluster-Bezeichnungen und die Skepsis der professionellen Patentrechercheure hinsichtlich des Einsatzes von Software-Tools, die „intelligente" Verarbeitungsmöglichkeiten versprechen.

4.1 Pre-Retrieval Clustering einer Kollektion

Die Grundlage für das Clustern von Dokumenten formuliert van Rijsbergen in seiner Cluster-Hypothese: „[. . .] closely associated documents tend to be relevant to the same requests." (van Rijsbergen 1979, 30) Diese Hypothese besagt, dass sich relevante Dokumente ähnlicher sind, als nicht-relevante Dokumente. In der Folge wurden daher zahlreiche Versuche unternommen, Clustering-Verfahren im Rahmen des IR zu integrieren. Panyr (1986, 87 f.) unterscheidet dabei drei Ansätze:

- Bei der **Dokumentenklassifikation** werden thematisch ähnliche Dokumente gruppiert. Motiviert wird dieser Ansatz aus Effizienzgründen: Im Vektorraummodell muss z.B. der Anfragevektor nicht mit allen Dokumentenvektoren verglichen werden, sondern nur mit den Cluster-Centroiden, was schneller zu bewerkstelligen ist.
- Die **Termklassifikation** soll eine Effektivitätssteigerung ermöglichen, indem thematisch ähnliche Terme gruppiert werden und diese ähnlichen Terme in einem darauf folgenden Retrievalprozess, z.B. zur Query-Expansion, eingesetzt werden können.
- Bei der **gleichzeitigen Term- und Dokumentenklassifikation** werden sowohl die Terme als auch die Dokumente automatisch gruppiert. Durch die gleichzeitige Anwendung des Clusterings auf beiden Ebenen erhofft sich Panyr in seinem eigenen Klassifikationsverfahren sowohl eine Effektivitäts-, als auch eine Effizienzverbesserung.

Eine Vielzahl von Experimenten wurde durchgeführt, um zu ermitteln, ob mittels Clustering-Verfahren die Retrieval-Ergebnisse insgesamt zu verbessern seien, wie es die Cluster-Hypothese erhoffen ließ (vgl. Hearst und Pedersen 1996, 77 f.). Es wurde dabei stets davon ausgegangen, dass im Sinne der Dokumentenklassifikation sämtliche Dokumente einer Kollektion statisch („persistent“ in der Terminologie von Maarek et al. (2002, 2)) im Vorfeld des eigentlichen Retrievals geclustert werden (daher der in dieser Arbeit verwendete Terminus *Pre-Retrieval Clustering*), ohne dabei Rücksicht auf eine konkrete Anfrage zu nehmen. In mehreren Experimenten konnte nachgewiesen werden, dass dieses Vorgehen nicht zu besseren Retrieval-Ergebnissen führt:

> „[...] retrieving the contents of the clusters whose centroids most closely match the query did not perform as well as retrieving the top ranked documents from the collection as a whole.“ (Hearst und Pedersen 1996, 77)

4.2 Post-Retrieval Clustering zur Aufbereitung von Ergebnismengen

Im vorangegangenen Kapitel wurde gezeigt, dass das Clustern von Dokumentenkollektionen im Vorfeld des Retrievals keine Verbesserung der Effektivität eines IR-Systems mit sich bringt. Daher wurde nach weiteren Anwendungsgebieten für die Clustering-Verfahren gesucht. Ein neues Gebiet wird im Clustern von Ergebnismengen gesehen, das im Verlauf dieses Kapitels vorgestellt wird.

4.2.1 Scatter/Gather-Ansatz

Den Vorschlag, Clustering-Verfahren auf Ergebnismengen von Suchanfragen anzuwenden, machten erstmals Cutting et al. mit ihrem Ansatz des Scatter/Gather: „Scatter/Gather may also be used to organize the results of word-based queries that retrieve too many documents.“ (Cutting et al. 1992, 319)

Maarek et al. (2002, 2) bezeichnen diese Art von Clustering im Gegensatz zum „persistent clustering“ (vgl. Kapitel 4.1) als „ephemeral clustering“ (engl. ephemeral = flüchtig, kurzlebig), um den temporären und dynamischen Charakter der Gruppenbildung zu kennzeichnen.

Der Ansatz von Scatter/Gather stellt eine Browsing-Methode dar: Beim Browsen (engl. = Stöbern) verschafft sich der Nutzer einen groben Überblick über den Inhalt eines Dokumentenkorpus. Zu vergleichen ist dies mit dem Stöbern im Inhaltsverzeichnis von Büchern, wodurch man z.B. auf interessante Abschnitte verwiesen wird und intensiver in den vorliegenden Text einsteigen kann. Im Gegensatz dazu

steht das zielgerichtete Search-Paradigma. Der Nutzer stellt eine konkrete Anfrage und ein System durchsucht das Dokumentenkorpus nach übereinstimmenden Dokumenten. In Analogie zur Buchmetapher entspricht dies dem Zugriff über ein Stichwortverzeichnis, um an die gewünschte Information zu gelangen. Der Nutzer muss jedoch seine Anfrage präzise formulieren und eventuell über das für den Gegenstandsbereich gängige Vokabular verfügen. Durch den Scatter/Gather-Ansatz soll ein Nutzer dabei unterstützt werden:

> „In particular, we anticipate that the browsing tool will not necessarily be used to find particular documents, but may instead help the user formulate a search request, which will then be serviced by some other means." (Cutting et al. 1992, 318)

Der Nutzer kann durch das Browsing zu neuen Ideen für die Formulierung seiner Suchanfrage gelangen und kann anschließend eine gezieltere Suche durchführen.

Das Prinzip von Scatter/Gather lässt sich wie folgt beschreiben: Das System verstreut (engl.: to scatter) die Kollektion in einzelne Cluster, die mit einer kurzen Zusammenfassung dem Nutzer präsentiert werden. Dieser wählt die ihn interessierenden Cluster aus, woraufhin die ausgewählten Cluster als Subkollektion zusammengetragen (engl.: to gather) werden. Auf die Subkollektion werden wiederum Clustering-Verfahren angewandt. Das Auswählen und erneute Clustern wiederholen sich solange, bis letztlich einzelne Dokumente angezeigt werden (vgl. Cutting et al. 1992, 319).

In der von Hearst und Pedersen (1996) durchgeführten Untersuchung gelangen die genannten Autoren zu dem Schluss, dass die Cluster-Hypothese auch für eine Ergebnismenge gilt, wobei der Kontext (festgelegt durch die Anfrage) eine entscheidende Rolle spielt:

> „[. . .] the clusters are created as function of which documents were retrieved in response to the query, and therefore have the potential to be more closely tailored to characteristics of a query than an independent, static clustering." (Hearst und Pedersen 1996, 78)

4.2.2 Clustern von Ergebnismengen im Web-IR

Zamir und Etzioni (1998) dehnten in ihrem Artikel den Anwendungsbereich von Clustering-Verfahren auf das Clustern von Web-Dokumenten aus. Dabei werden anhand der kurzen Beschreibungen, die von Suchmaschinen als Ergebnis zurückgeliefert werden, Cluster von Dokumenten ermittelt, die dem Nutzer die Navigation in den Suchergebnissen erleichtern sollen. Dieses Vorgehen wird in dieser Arbeit

mit *Post-Retrieval Clustering* bezeichnet. In der Meta-Suchmaschine *MetaCrawler*[1] wurde das für diesen Zweck von Zamir und Etzioni entwickelte Verfahren Suffix-Tree-Clustering kommerziell umgesetzt. Daneben gibt es weitere Suchmaschinen im World Wide Web, die ein Clustern der Suchergebnisse ermöglichen, wie z.B. Vivisimo[2] oder neuerdings die Suchfunktion von Web.de[3].

Besondere Anforderungen an die Clustering-Verfahren werden durch den Charakter des Web-IR gestellt (zitiert nach Zamir und Etzioni 1998, 46):

1. **Relevance:** The method ought to produce clusters that group documents relevant to the user's query separately from irrelevant ones.
2. **Browsable Summaries:** The user needs to determine at a glance whether a cluster's contents are of interest. We do not want to replace sifting through ranked lists with sifting through clusters. Therefore the method has to provide concise and accurate descriptions of the clusters.
3. **Overlap:** Since documents have multiple topics, it is important to avoid confining each document to only one cluster.
4. **Snippet-tolerance:** The method ought to produce high quality clusters even when it only has access to the snippets returned by the search engines, as most users are unwilling to wait while the system downloads the original documents off the Web.
5. **Speed:** A very patient user might sift through 100 documents in a ranked list presentation. We want clustering to allow the user to browse through at least an order of magnitude more documents. Therefore the clustering method ought to be able to cluster up to one thousand snippets in a few seconds. For the impatient user, each second counts.
6. **Incrementality:** To save time, the method should start to process each snippet as soon as it is received over the Web.

Zamir und Etzioni vergleichen in einem zuvor veröffentlichten Artikel (vgl. Zamir und Etzioni 1998, 52), ob es einen Unterschied macht, wenn nur die von den Suchmaschinen gelieferten Informationen oder das gesamte Web-Dokument als Ausgangsmaterial für das Clustering verwendet werden. Der Verlust an Qualität sei relativ gering, obwohl im Volltext 760 Terme (220 Terme nach Entfernen von Stoppwörtern) und in den Suchmaschinenergebnissen nur 50 Terme (20 Terme ohne Stoppwörter) vorliegen. Als Erklärung für dieses Verhalten vermuten die Autoren, dass Suchmaschinen versuchen, nur bedeutungstragende Phrasen zu extrahieren. Das verringere das „Rauschen" in den Daten gegenüber den Volltexten, was sich positiv auf das Gesamtergebnis auswirke.

[1] http://www.metacrawler.com, Verifizierungsdatum: 12.11.2004, 23:55 Uhr MEZ
[2] http://www.vivisimo.com, Verifizierungsdatum: 12.11.2004, 23:55 Uhr MEZ
[3] http://www.web.de → Suche, Verifizierungsdatum: 12.11.2004, 23:55 Uhr MEZ

Die oben von Zamir und Etzioni formulierten Anforderungen lassen sich größtenteils auf den Einsatz von Clustering-Verfahren im Anwendungsbereich der Patentrecherche übertragen:

ad 1.) Die automatisch erstellten Cluster, die auf den zurückgelieferten Ergebnissen einer Suchanfrage an eine Patentdatenbank basieren, sollen zu thematisch kohärenten Gruppen zusammengefasst werden (was den relevanten Dokumenten entspricht).

ad 2.) Die automatisch erstellten Cluster sollen mit einer geeigneten Beschreibung zur Kennzeichnung des Clusterinhalts versehen werden, damit der Nutzer zwischen relevanten und nicht-relevanten Clustern schnell unterscheiden kann.

ad 3.) Wie bei der Klassifikation von Patentschriften nach der IPC, bei der zur Kategorisierung eine Hauptklasse und eventuell mehrere Nebenklassen vergeben werden können, so sollen die Patentdokumente nicht nur einem Cluster, sondern mehreren Clustern gleichzeitig angehören können. Diese Anforderung würde ein Clustering-Verfahren voraussetzen, dass graduelle Zugehörigkeiten zu einem Cluster ermitteln könnte sowie einen Schwellenwert, der bestimmt, ab welchem Grad ein Dokument in einem Cluster erscheint. Konzeptuell wird dies von probabilistischen oder fuzzy-Clustering Algorithmen ermöglicht (siehe hierzu Kapitel 7.3 und 7.5.1).

ad 4.) Die Forderung nach einer *Snippet-Tolerance* berührt die Frage, welche Datengrundlage angesichts des Anwendungsbereiches angemessen ist und die wahrscheinlich am besten experimentell zu beantworten wäre: Reichen die Informationen der Datenbank PATDPA aus oder müssen eher die Volltexte der Patentschriften aus der Datenbank PATDPAFULL herangezogen werden (siehe Kapitel 3.2.4)?

ad 5.) Die eingesetzten Verfahren zur Clusterbildung müssen hinsichtlich des Datenaufkommens gut skalieren und in angemessener Zeit Resultate errechnen, was vor allem im Online-Betrieb von großer Bedeutung ist. Würden sich eventuell Volltexte als geeignetere Datengrundlage herausstellen, hätte dies große Auswirkungen auf den Verarbeitungsaufwand, was die Antwortzeit des Systems sicherlich verlängern würde, bis der Nutzer seine geclusterten Ergebnisse präsentiert bekäme.

ad 6.) Beim Clustern von Patentdokumenten aus einer Online-Datenbank bestehen andere Voraussetzungen als beim Clustern von Web-Dokumenten. Web-Dokumente sind in der Regel verstreut auf verschiedenen Servern gespeichert und müssen erst über das World Wide Web angefordert und übertragen werden, um sie weiterzuverarbeiten. Dahingegen stehen Online-Datenbanken meist zentral auf Großrechnern eines Hosts zur Verfügung, über den auch die Anfragen bearbeitet werden. Dies ermöglicht einen effizienten Zugriff auf die dort direkt gespeicherten Daten, ohne dass ein vorhergehendes „Einsammeln“ der Dokumente, wie zuvor beim Web-IR beschrieben, anfällt.

4.2.3 Automatisches Bezeichnen von Clustern

Zamir und Etzioni (1998) stellten in ihren Anforderungen an das Web-IR (siehe vorheriger Abschnitt) die Forderung auf, dass „browsable summaries“ existieren sollen, die einen Clusterinhalt schlüssig bezeichnen. Verschiedene Ansätze existieren, um Cluster automatisch zu bezeichnen, die im Folgenden kurz vorgestellt werden.

Häufig vorkommende Terme

Oftmals werden dazu die am häufigsten vorkommenden Terme verwendet. Die für die Experimente im Rahmen dieser Arbeit eingesetzt Software CLUTO ermittelt die Bezeichnungen dadurch, dass die Terme ausgewählt werden, die „contribute the most to the average similarity between the objects of each cluster.“ (Karypis 2003, 16) Da jedoch die Terme im Rahmen der Experimente in gestemmter Form vorliegen, werden sie auch so ausgegeben. Die dadurch erzeugten Beschreibungen lauten wie folgt: „information, ueb, anwend“ oder „elektron, uebertrag, comput“. Insgesamt sind diese Beschreibungen nicht leicht lesbar und nicht einfach verständlich, was angesichts der Forderung von Zamir und Etzioni wünschenswert gewesen wäre.

Popescul und Ungar üben Kritik an der Herangehensweise, die häufigsten Terme zur Beschreibung zu verwenden, da

> „The lists of the most frequent words often reveal the topic at a high level, but can fail to depict cluster-specific details as they are diluted with what we call *collection specific stop words*. E.g., in a collection of computer science research papers, terms such as *paper, method, result, system, or present* are very frequent and are common to most computer science subdisciplines, therefore giving no additional information to someone who already knows that all of the documents are computer science research papers.“ (Popescul und Ungar 2000, 2)

In ihrem Artikel gelangen sie auf Basis eines kleinen Nutzertests zu dem Ergebnis, das Clusterbeschreibungen, die mit Hilfe der „most frequent and most predictive“ Termen gewonnen wurden, die aussagekräftigsten seien (vgl. Popescul und Ungar 2000, 14). Dabei wird ähnlich der TF-IDF Gewichtung im IR vorgegangen: Terme, die in der gesamten Kollektion häufig vorkommen, erhalten ein geringes Gewicht während Termen, die häufig innerhalb eines Clusters erscheinen, ein größeres Gewicht zuteil wird. Formal: Sei $p(\text{word}|\text{class})$ die Frequenz eines Terms innerhalb eines Clusters und $p(\text{word})$ die Frequenz eines Terms innerhalb der Kollektion, dann errechnet sich das Gewicht eines Terms aus (Yarowski, zitiert nach Popescul und Ungar 2000, 4 f.):

$$p(\text{word}|\text{class}) * \frac{p(\text{word}|\text{class})}{p(\text{word})}$$

Lexical Affinities

Maarek et al. (2002) verfolgen einen anderen Ansatz, um die Qualität der Beschreibungen zu erhöhen:

> „Instead of single words as indexing units, our indexing unit consists of a pair of words that are linked by a *lexical affinity* (LA). An LA between two units of language stands for a correlation of their common appearance.“ Maarek et al. (2002, 20)

Single words	Lexical Affinities
0.37 merced	0.20 county*merced
0.29 yosemite	0.13 national*park
0.12 county	0.13 national*yosemite
0.12 hour	0.13 park*yosemite
0.08 populate	0.08 drive*hour
...	...

Tabelle 4.1: Einzelterme im Vergleich mit LA-Termen zur Inhaltsbezeichnung für die Web-Site “Merced County“

Für die Web-Seite „Merced County“[4], die ein Resultat auf die Suchanfrage „merced“ an die Suchmaschine Google ist, verglichen Maarek et al. (2002, 8) die termbasierte mit der LA-basierten Indexierung (siehe Tabelle 4.1). Hinsichtlich des vierten Elements „hour“ (Spalte „single words“ in Tabelle 4.1) kann sich ein Nutzer fragen, wie dies mit der Web-Site zusammenhängt. Das fünfte Element der „Lexical Affinities“ („drive*hour“) lässt eher Rückschlüsse auf den Inhalt zu. Wahrscheinlich handelt es sich um Fahrzeiten, die in „drive hours“ angegeben werden.

Phrase Intersection Clustering

Zamir et al. (1997) verwenden einen ähnlichen Ansatz, den sie *phrase intersection clustering* nennen: „This approach treats a document as a sequence of words, with the premise that phrases found in the document can be useful both for the clustering algorithm and as an indication of the cluster’s content.“ (Zamir et al. 1997, 4) Beispielsweise werden dann auf die Suchanfrage „Clinton“ Phrasen zurückgeliefert, die in vielen Dokumenten gemeinsam vorkommen, z.B. „democratic party“ und „Hillary Rodham Clinton“, um so dem Nutzer ein besseres Bild vom Inhalt eines Clusters zu geben.

[4] Zum Zeitpunkt der Publikation des Artikels (2002) unter `http://www.co.merced.ca.us/About_us/index.html` zu erreichen. Die Seite ist inzwischen verschoben worden und ist jetzt aktuell unter `http://www.co.merced.ca.us/CountyWeb/pages/linked.aspx?path=general/aboutus.html` zu erreichen.

Implikationen für das Vorgehen in dieser Arbeit

Für die im Zuge dieser Arbeit durchgeführten Clustering-Experimente (siehe Kapitel 8.4) erfolgt keine automatische Bezeichnung der Clusterinhalte. Dies hat vor allem praktische Gründe, da nur die Software CLUTO eine automatische Benennung unterstützt. Will man gezielt andere Verfahren zur Generierung einer Benennung einsetzen, so muss man Änderungen im Quellcode vornehmen, der z.B. im Falle von CLUTO nicht veröffentlicht ist. Eine weitere Möglichkeit, um die in diesem Kapitel vorgestellten Verfahren zur Erzeugung einer Benennung einzusetzen, besteht darin, die Clustering-Algorithmen in eigenen Implementationen umzusetzen, was insgesamt sehr aufwändig, fehlerträchtig und oftmals nicht so performant wie die vorhandenen Implementationen zu leisten ist.

Da sich dieser Teilbereich insgesamt als sehr komplex und umfangreich darstellt, und der Schwerpunkt dieser Arbeit auf der Analyse vorhandener Clustering-Algorithmen liegt, erfolgt keine weitere Bearbeitung dieses Themenbereichs im Rahmen dieser Arbeit.

4.3 Kritik an der Darstellung von Ergebnismengen als Cluster

Im Rahmen seiner Doktorarbeit führte Kural eine Nutzerbefragung durch, um zu ermitteln, welche Darstellungsart effektiver ist: geclusterte Ergebnismengen oder nach Relevanz sortierte Listen (vgl. Kural et al. 2001, 594). Die Versuche basierten auf der Annahme, dass Nutzer nur an ein oder zwei Aspekten der Ergebnismenge interessiert seien, und nicht an einem groben Überblick über die vorhandenen Themen innerhalb einer Anfrage (vgl. Kural et al. 1999, 7). Als Ergebnis wurde formuliert:

> „Clustering did not appear to be preferable to ranked lists especially as it also represented overheads in both computing time and resources involved in creation of the clusters . . . “ (Kural et al. 2001, 596)

Den Nutzern gelänge es eher, nicht-relevante Cluster zu identifizieren (in 37% der Fälle) und das, so die Autoren, widerspricht den von Hearst und Pedersen (1996) formulierten Aussagen, dass Nutzer die Cluster von höchster Precision eindeutig ermitteln könnten (in den Experimenten von Kural gelang dies nur 29% der Nutzer). Per Fragebogen wurden die Nutzer anschließend zu ihren Erfahrungen mit dem System befragt, woraus Kural et al. (2001, 597 ff.) folgende Hauptkritikpunkte zusammenfassen:

- *„Cluster representations are not always adequately informative [. . .]“*
  Der hohe Informationsverlust, der beim Verdichten des Clusterinhalts auf ein oder mehrere Schlagworte auftritt, wird als Grund dafür angegeben. Kural et

al. führen dies darauf zurück, dass eine zweite Abstraktionsschicht zwischen den Nutzer und den Dokumenten eingefügt wird. Dieser Effekt sei besonders gravierend, da die Titel- oder Abstract-Informationen, die den Nutzern gezeigt werden, bereits eine Zusammenfassung des eigentlichen Dokuments darstellen: „Cluster representation is a representation of representations." (Kural et al. 2001, 598)

- *„Cluster representations may be misleading."*
  Einige Nutzer fühlten sich durch die Cluster-Beschreibung getäuscht und vermuteten keine relevanten Dokumente innerhalb dieser Cluster, obwohl welche vorhanden waren.
- *„Users' own perceptions about document similarities may not be reflected in the grouping provided by the algorithm."*
  Die automatisch erstellten Cluster ähneln nicht unbedingt den erwarteten Gruppierungen der Nutzer. Jedoch unterscheiden sich die von unterschiedlichen Nutzern manuell erstellten Clustereinteilungen stark voneinander, was in einer anderen Untersuchung von Macskassy et al. (1998) aufgezeigt wurde:

  > „Each subject tended to be diverse in his or her clustering across the five queries and little similarity was found between different subjects." (Macskassy et al. 1998, 10)

- *„Users' expectations from the clustering may exceed what the clusters can offer."*
  Die einen Nutzer erwarten nur wenige Cluster mit 100% Precision, andere wiederum erwarten, dass sämtliche erzeugten Cluster eine hohe Precision aufweisen. Eine weitere Nutzergruppe erwartet hingegen, dass anhand der Cluster-Bezeichnung eindeutig voneinander getrennte Cluster unterschieden werden können (vgl. Anick und Vaithyanathan, zitiert nach Kural et al. (2001, 600)).

Die Darstellung dieser Beobachtungen soll verdeutlichen, dass Clustering-Lösungen nicht *per se* von den Nutzern als eine bessere Art der Präsentation von Ergebnissen aufgefasst werden. Clustering-Lösungen müssen daher einen eindeutigen Mehrwert aus Nutzersicht aufweisen, damit eine ausreichende Akzeptanz erzielt wird. Dies gilt insbesondere für den Anwendungsbereich der Patentrecherche und Patentinformation, worauf im folgenden Kapitel eingegangen wird.

4.4 Clustering-Verfahren als Werkzeuge zur Patentanalyse und -recherche

Im Anwendungsbereich der Patentrecherche, der dieser Arbeit zu Grunde liegt, bietet sich der Einsatz von Clustering-Verfahren an. Im Folgenden werden zunächst bestehende Ansätze vorgestellt und anschließend deren Einsatz aus Sicht von Experten und Laien bewertet.

Abbildung 4.1: Darstellung einer Clustering-Lösung durch ThemeScape, Quelle: http://www.researchinformation.info/rijanfeb04patent1.html

4.4.1 Patinformatics und Text Mining als „Werkzeuglieferanten“

Die Analyse von Patentdokumenten zur Ermittlung von Beziehungen und Trends wird von Trippe mit dem Begriff *Patinformatics* bezeichnet:

> „[...] the term patinformatics describes the science of analyzing patent information to discover relationships and trends, which would be difficult to see when working with patent documents on a one-to-one basis.“ (Trippe 2003, 211)

Trippe (2003, 213) stellt in seinem Artikel Techniken und Software-Lösungen vor, die im Rahmen der „Patinformatics“ genutzt werden. Er unterscheidet beim Clustering von Patentdokumenten zwischen

1. dem Clustern von strukturierten Daten (Feldern der Datenbank), um Dokumente zu gruppieren, die eine ähnliche Struktur der Datenfelder aufweisen.
2. dem Clustern von unstrukturierten Daten (Texten), um Dokumente mit ähnlichen Konzepten zu gruppieren.
3. dem „Patent Mapping“, um Beziehungen zwischen Clustern zu identifizieren. Hierbei werden Dokument-Cluster zweidimensional in Form einer Karte visualisiert, so dass Gruppen von ähnlichen Dokumenten (optisch) nahe beieinander liegen (siehe Abbildung 4.1).

Das Clustern von Dokumenten zur Strukturermittlung in Dokumentenkollektionen wird beim *Text Mining* als ein Teilgebiet innerhalb dieser Disziplin betrachtet. Das

Aufgabengebiet von Text Mining wird mit der „automatischen Erschließung von Textinhalten und -zusammenhängen“ umschrieben (Gerstl et al. 2001, 38). Dazu werden dem Data Mining ähnliche Verfahren angewandt, um in den unstrukturierten, natürlichsprachlichen Texten unbekannte Muster und Zusammenhänge zu entdecken.

Im folgenden Abschnitt werden die Herangehensweisen von Experten und Nicht-Experten bei einer Patentrecherche beschrieben und Anknüpfungspunkte für die Anwendung von Clustering-Verfahren aufgezeigt.

4.4.2 Ablauf einer Recherche und Einbindung neuer Werkzeuge zur Analyse von Patentdokumenten

Ein Informationssuchender formuliert während eines Retrievalprozesses zunächst seinen Informationsbedarf in Form einer Suchanfrage (in diesem Fall sucht er nach Patentdokumenten). Als Resultat auf seine Anfrage an eine Patentdatenbank erhält er eine Antwortmenge von Patentdokumenten. Diese Menge soll nun durch Text-Mining Verfahren, genauer gesagt durch Clustering-Verfahren, aufbereitet werden, so dass für den Nutzer ein Mehrwert entsteht (z.B. durch Ermittlung von Trends als mögliche Erkenntnis aus einer durchgeführten Gruppierung). Jedoch stehen professionelle Patent-Rechercheure diesen Werkzeugen kritisch gegenüber:

> „[...] within the professional patent information community there still is a high degree of scepticism as regards the use of these new linguistic technologies. At least in part, this is due to the relative ‚black box’ effect inherently attached to the nature of the said technology.“ „[...] professional patent searchers are rather suspicious of tools that do not generally grant the user complete control over their inner workings.“ (Fattori et al. 2003, 335)

Um Patentdokumente oder bibliographische Informationen zu recherchieren, bevorzugen Experten auf Grund der subjektiv größeren Kontrolle über das Suchergebnis das klassische Boolesche Retrieval. Als Experten beherrschen sie Verknüpfungsoperatoren wie z.B. AND und OR und können komplexe Suchanfragen mittels weiterer Operatoren und Klammern formulieren. Für Laien, die eine Patentdatenbank nutzen, die ausschließlich Boolesches Retrieval unterstützt, ist diese Art der Anfrageformulierung ungeeignet, da ein Verständnis der Booleschen Logik vorliegen muss.

Allgemein ist bei Booleschen Anfragen der Umfang der Ergebnismenge schwer zu kontrollieren: Entweder erhält man zu viele Ergebnisse oder zu wenige, weil beispielsweise die Suchanfrage zu eng gefasst wurde. Zudem wird die gesamte Dokumentenkollektion in relevant bzw. nicht relevant unterteilt, so dass in der zurückgelieferten Ergebnismenge alle Dokumente gleich „wichtig“ erscheinen. Dies ermöglicht keine graduelle Abstufung z.B. in Form eines Rankings, durch die ein Nutzer die

(wahrscheinlich) relevantesten Dokumente zuerst präsentiert bekommt (vgl. Cooper 1988).

Clustering-Verfahren, die die Ergebnismenge auf eine Anfrage vorsortieren, können sowohl für Gelegenheitsnutzer und Laien, als auch für professionelle Rechercheure eine Hilfe darstellen. Falls diese ihre Suchanfragen zu weit gefasst formuliert haben, können im Idealfall zusammengehörige Gruppen erzeugt werden, so dass ein Nutzer gezielt in diesen Clustern suchen kann oder Anregungen zum Umformulieren der ursprünglichen Suchanfrage erhält. Für die Gruppe der Experten wären Eingriffsmöglichkeiten zu integrieren, damit sie auf die „Black-Box“ Clustering-Verfahren einwirken und somit das Ergebnis beeinflussen können, z.B. wie in dem von Fattori et al. (2003, 336) als Prototyp realisierten System durch Festlegen der Clusteranzahl, des Termgewichtungsschemas und anderer Parameter.

4.5 Zusammenfassung

Dieses Kapitel widmete sich den bestehenden Einsatzmöglichkeiten von Clustering-Verfahren. Bislang wurde in den hierzu veröffentlichten Publikationen versucht, durch Pre-Retrieval Clustering einer Dokumentenkollektion Vorteile für das Retrieval von Dokumenten zu erhalten. Nachdem sich dieser Ansatz als nicht erfolgreich herausgestellt hat, gewann in den vergangenen Jahren zunehmend das Post-Retrieval Clustering zur Aufbereitung der Ergebnismengen einer Suchanfrage an Bedeutung. Trotz kritischer Stimmen, die die Eignung dieser Art von Ergebnispräsentation anzweifeln, werden für das Post-Retrieval Clustering beständig neue Anwendungsgebiete erschlossen. Dies trifft auch auf das Gebiet der Patentrecherche zu, das den Anwendungsbereich dieser Arbeit darstellt. Jedoch muss den Eigenheiten dieses Bereichs, z.B. der Skepsis der professionellen Rechercheure gegenüber „Black-Box“-Tools, besonders Rechnung getragen werden.

Die folgenden Kapitel gliedern sich anhand des Ablaufs einer Cluster-Analyse (vorgestellt in Kapitel 2.1) und beschreiben Station für Station, unter Berücksichtigung des Anwendungsgebiets Patentrecherche, die zu tätigenden Überlegungen und Maßnahmen von der Aufbereitung der Daten bis hin zur Interpretation der Ergebnisse.

5 Auswahl und Aufbereitung der Attribute

Die Ausgangsdaten für die Experimente in der vorliegenden Arbeit bestehen aus Patentdokumenten oder, genau genommen, aus den Termen und der Häufigkeit ihres Vorkommens innerhalb der Patentdokumente. Ein Dokument kann man sich anschaulich als ein „bag of words“ vorstellen. Dieser Vorstellung eines Dokuments liegt eine ungeordnete Menge von Wörtern zu Grunde, wobei in dieser Menge ein Element mehrfach vorkommen kann. Im Folgenden werden Grundlagen zur Attributaufbereitung aufgezeigt, die im Rahmen des Clusterns von Dokumenten von Bedeutung sind.

5.1 Vektorraummodell und Clustering von Dokumenten

Die Ausgangsdaten werden in Form einer $n \times m$ Datenmatrix (siehe Abbildung 5.1) zur Weiterverarbeitung für die Folgeschritte der Clusteranalyse (Proximitätsberechnung – Fusionierungsschritt) bereitgestellt. Die Zeilen bezeichnen die m Objekte, was den einzelnen Patentdokumenten entspricht. Die Spalten beschreiben die n Attribute eines Objekts, d.h. welche Terme in einem Patentdokument vorkommen.

$$\begin{pmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{32} & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{m1} & d_{m2} & d_{m3} & \cdots & d_{mn} \end{pmatrix}$$

Abbildung 5.1: Ausgangsdaten als Datenmatrix

Diese Sichtweise der Matrixschreibweise gleicht dem im Information Retrieval weit verbreiteten Vektorraummodell, das in Arbeiten von Gerald Salton 1971 im Rahmen des SMART-Retrieval Systems an der amerikanischen Universität Cornell erstmals formuliert wurde (vgl. Belew (2000, 86) und vgl. Womser-Hacker (2003, 1)). Das Vektorraummodell wird formal folgendermaßen definiert:

> „Sei $D = d_1, \ldots, d_m$ eine Menge von Dokumenten oder Objekten und $A = A_1, \ldots, A_n$ eine Menge von Attributen $A_j : D \to \mathcal{R}$ auf diesen Objekten. Die Attributwerte $A_j(d_i) =: w_{i,j}$ des Dokuments d_i lassen sich als Gewichte auffassen und zu einem Vektor $w_i = (w_{i,1}, \ldots, w_{i,n}) \in \Re^n$ zusammenfassen. Dieser Vektor beschreibt das Dokument im Vektorraummodell: Er ist seine Repräsentation und wird *Dokumentenvektor* genannt.“ (Ferber 2003, 63)

5.2 Attributtypen

Ein Objekt wird durch seine Attribute oder Merkmale beschrieben, deren Ausprägungen mit Hilfe einer Skala gemessen werden. Je nachdem, welche Art der Messung an einem Attribut möglich ist, verfügt eine Skala über ein bestimmtes *Skalenniveau bzw. einen Skalentyp*. Die Art des Skalentyps entscheidet darüber, welche Proximitätsmaße bzw. Fusionierungsalgorithmen zur Clusterbildung direkt oder eventuell erst aber nach einer Umformung der Merkmale angewandt werden können. Tabelle 5.1 liefert einen Überblick über die verschiedenen Skalen und deren Anwendung (siehe hierzu auch Backhaus et al. 2003, 4 ff.).

Da im Vektorraummodell die gewichteten bzw. ungewichteten Termfrequenzen als Attribute verwendet werden, kann man den Skalentyp als ratio-skaliert bezeichnen: Die Attribute weisen eine quantitative Ausprägung auf, wobei das Vorhandensein eines Terms mit einer Termfrequenz $\text{tf} > 0$ und das Nicht-Vorhandensein mit $\text{tf} = 0$ als „Nullpunkt“ der Skala beschrieben werden kann.

qualitativ (nicht-metrische Skalen)	**nominal**	Dies sind Namen oder Bezeichnungen, die qualitative Eigenschaften kennzeichnen. *Beispiele:* Farbe (rot - gelb - grün - blau ...) oder Geschlecht (männlich - weiblich)
	ordinal	Es kann eine Rangordnung erstellt werden. Die Werte sagen aber nichts über die Abstände zwischen den Objekten aus. *Beispiel:* Schulnoten (1, 2, 3, 4, 5, 6), Lautstärke (laut - leise) oder Geschmack (gut - besser - am besten)
quantitativ (metrische Skalen)	**intervall**	Diese Skala weist gleich große Skalenabschnitte auf. Unterschiede zwischen den Werten sind von Bedeutung (z.B. als Differenz). *Beispiel:* Celsius-Skala
	ratio	Es existiert ein natürlicher Nullpunkt, so dass man bei „0“ sagen kann: „Merkmal nicht vorhanden“. *Beispiele:* Gewicht, Länge, Geschwindigkeit, Einnahmen, Preis.

Tabelle 5.1: Verschiedene Skalen und ihre Eigenschaften

5.3 Gewichtung der Terme

Das Gewicht eines Terms kann entweder durch simples Aufsummieren seines Vorkommens in einem Dokument oder durch effektivere Verfahren ermittelt werden. Ziel einer Gewichtung ist es, Terme (bzw. Deskriptoren) zu identifizieren, die ein Dokument von anderen Dokumenten gut diskriminieren. Zur Berechnung der Gewichte werden folgende Angaben benötigt:

Größe	Symbol	Definition
Termfrequenz (term frequency)	$\text{tf}_{i,j}$	Anzahl des Vorkommens von Term_i in Dokument d_j
Dokumentfrequenz (document frequency)	df_i	Anzahl der Dokumente innerhalb der Kollektion, die Term_i enthalten
Kollektionsfrequenz (collection frequency)	cf_i	Gesamtzahl des Vorkommens von Term_i in einer Kollektion

Tabelle 5.2: Größen zur Termgewichtung (vgl. Manning und Schütze 2002, 542)

Bei der Berechnung der Termfrequenz geht man davon aus, dass Terme, die häufig in einem Dokument auftreten, den Inhalt am ehesten beschreiben. Das stellt eine lokale Gewichtung auf Dokumenten-Ebene dar. Die Dokumentfrequenz charakterisiert die Aussagekraft eines Terms global über eine Kollektion hinweg. Kommt ein Term sehr häufig in einer Kollektion vor, so ist er nicht besonders spezifisch. Ein Term, der in einer Kollektion nicht sehr oft vorkommt, ist dagegen spezifischer und grenzt den Inhalt eines Dokuments stärker ein, weshalb er ein geeigneterer Index-Term ist. Lokale und globale Gewichtungsmethoden werden häufig kombiniert, um ein Gesamt-Gewicht zu ermitteln, darunter die beiden im Weiteren exemplarisch vorgestellten Gewichtungs-Schemata (vgl. Ferber 2003, 66 ff.).

5.3.1 Gewichtung nach TF/IDF

Bei der so genannten TF/IDF-Gewichtung wird die Termfrequenz (tf) und die inverse Dokumentfrequenz (idf) miteinander verknüpft. Dabei werden Terme, die häufig innerhalb eines Dokuments, aber selten innerhalb einer Kollektion vorkommen, besonders stark gewichtet. Kommt ein Term in einem Dokument nicht vor, erhält er das Gewicht 0. Eine Möglichkeit, Terme mittels TF/IDF zu gewichten, ist nachfolgend dargestellt (vgl. Manning und Schütze 2002, 543 f.):

$$w_{(i,j)} = \begin{cases} (1 + \log(\text{tf}_{i,j})) * log\frac{N}{\text{df}_i} & \text{falls} \quad \text{tf}_{i,j} \geq 1 \\ 0 & \text{falls} \quad \text{tf}_{i,j} = 0 \end{cases}$$

5.3.2 Gewichtung nach Okapi-BM25

Bei der Evaluation von IR-System im Rahmen von TREC[1] 1–7 sicherte sich das Okapi-System[2] mit seiner Art der Termgewichtung wiederholt Ergebnisse in der Spitzengruppe. Innerhalb des Systems sind verschiedene Gewichtungs-Schemata realisiert,

[1] Text Retrieval Conference, internationale Konferenz zur Evaluierung von IR-Systemen

[2] Das Okapi Basic Search System (Okapi BSS) ist mittlerweile fest in die vom Microsoft Research Labor in Cambridge entwickelte Umgebung *Keenbow* integriert worden, die als Framework zur Evaluation verschiedener IR-Verfahren dient (Robertson und Walker 2000, 2).

von denen sich das „Best Match 25“ (BM 25) Verfahren am effizientesten erwies (vgl. Robertson und Walker 2000, 1 f.).

Die Gewichtungs-Formel, die seit TREC-3 im Wesentlichen unverändert geblieben ist, verwendet die in Kapitel 5.3 eingeführten lokalen (tf) und globalen (idf) Maße und ergänzt diese zusätzlich um eine Normierung anhand der Dokumentlänge. Dadurch wird erreicht, dass Terme aus längeren Dokumenten nicht automatisch ein größeres Gewicht erhalten als Terme aus kürzeren Dokumenten. Nachfolgend wird eine im Vergleich zum Original leicht vereinfachte Fassung der BM25-Formel wiedergegeben, die so auch für die im Zuge dieser Arbeit durchgeführten Experimente (Kapitel 8) eingesetzt wurde (nach Robertson et al. 2000, 96 f.):

$$\sum_{T \in \mathcal{L}} w^{(1)} \frac{(k_1 + 1) * \mathrm{tf}}{K + \mathrm{tf}} * \mathrm{qtf}$$

$w^{(1)}$ entspricht der Robertson-Sparck-Jones-Formel zur Gewichtung eines Terms in Abhängigkeit einer Anfrage und ist definiert durch:

$$w^{(1)} = \log \frac{(r + 0.5)/(R - r + 0.5)}{(n - r + 0.5)/(N - n - R + 0.5)}$$

Dabei gilt: n entspricht der Anzahl der Dokumente, die den Term enthalten; R entspricht der Anzahl der relevanten Dokumente innerhalb einer Trainingsmenge; r entspricht der Anzahl der relevanten Dokumente, die den Term enthalten. Da jedoch keine Relevanzinformationen vorliegen, liefert diese Formel ein Gewicht ähnlich einer Gewichtung nach IDF.

Weitere Parameter der Okapi BM25 Gewichtung sind $K = k_1((1-b)+b*\mathrm{dl}/\mathrm{avdl})$, wobei k_1 und b frei wählbare Parameter sind, die je nach Art der Kollektion experimentell ermittelt werden müssen; dl ist die Länge eines Dokuments und avdl entspricht der durchschnittlichen Länge eines Dokuments (gemessen in einer geeigneten Einheit, z.B. Anzahl der Terme). tf steht für die Termfrequenz ($\mathrm{tf}_{i,j}$) und qtf für die Dokumentfrequenz ($\mathrm{df}_{i,j}$).

5.4 Standardisierung bzw. Normierung von Attributen

Je nach verwendetem Proximitätsmaß wird von verschiedenen Autoren eine Standardisierung bzw. Normierung der Rohdaten in der Datenmatrix empfohlen. Sie befürchten eine Verzerrung des Clustering-Ergebnisses, da z.B. bei großen Unterschieden im Absolutwert oder in der Varianz von Attributwerten die Variablen mit dem größeren Wert einen stärkeren Einfluss ausüben (vgl. Milligan 1996, 352).

In der multivariaten Analyse wird klassischerweise die *Z-Transformation* als Standardisierungsmethode vorgestellt, die in ihrer generellen Fassung wie folgt lautet (vgl. Kaufmann und Pape 1984, 283):

$$\widetilde{x}_{ni} = \frac{x_{ni} - \overline{i}_j}{s_i^q} \qquad n = 1, \ldots, N; \quad i = 1, \ldots, p \tag{5.1}$$

mit dem Mittelwert des Merkmals i

$$\overline{x}_i = \frac{1}{N} \sum_{n=1}^{N} x_{ni} \qquad i = 1, \ldots, p \tag{5.2}$$

und

$$s_i^q = \left(\frac{1}{N} \sum_{i=1}^{N} |x_{ik} - \overline{x}_k| \right)^{1/q} \qquad i = 1, \ldots, p. \tag{5.3}$$

Wählt man in Gleichung 5.3 für den Parameter $q = 2$, so erhält man die Standardabweichung, die bei der z-Transformation dafür sorgt, dass alle Merkmale „einen Mittelwert von Null und eine Varianz von Eins besitzen [...]“ (Backhaus et al. 2003, 539). Für den Parameter $q = 1$ in Gleichung 5.3 bemerken Hösel und Walcher, dass dieses Streuungsmaß nicht in dem Maße durch Ausreißer beeinflusst wird, wie dies bei der Standardabweichung geschieht (vgl. Hösel und Walcher, 7).

Milligan (1996, 352) kritisiert, dass eine Standardisierung die Cluster, die in den Ausgangsdaten vorhanden sind, möglicherweise verfälsche oder sogar zerstöre. Sie sei nur dann gerechtfertigt, wenn nach der Standardisierung die Cluster erhalten blieben. Zudem sei die Z-Transformation nicht immer die geeignetste Methode. In einem Vergleich verschiedener Standardisierungsverfahren anhand von synthetischen Daten erwiesen sich die Verfahren am erfolgreichsten, die die Spannweite der Variablen (siehe Nenner) einbeziehen:

$$z_4 = \frac{x}{\max(x) - \min(x)} \quad \text{oder} \quad z_5 = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Auf Grund der Gefahr, dass Cluster durch eine Standardisierung verzerrt werden können, werden die Attribute für die im Rahmen dieser Arbeit durchgeführten Experimente (siehe Kapitel 8) keiner Normierung bzw. Standardisierung unterzogen.

5.5 Zusammenfassung

In diesem Kapitel wurde die Auswahl und Aufbereitung der Attribute beschrieben. Die Patentdokumente, die die Datengrundlage für die durchgeführten Experimente im praktischen Teil dieser Arbeit bilden, werden im Vektorraummodell repräsentiert. Die dafür verwendeten Attributtypen sind ratio-skaliert, da Termhäufigkeiten als Attributwerte verwendet werden. Außerdem wurden zwei Möglichkeiten zur Termgewichtung vorgestellt, von denen die Gewichtung nach der Okapi BM25 für die Experimente (siehe Kapitel 8.4) eingesetzt wurde. Im Weiteren wurden in diesem Kapitel Verfahren zur Standardisierung bzw. Normierung von Attributwerten präsentiert, deren Anwendung im Rahmen einer Cluster-Analyse eher skeptisch beurteilt wird.

6 Proximitätsmaße

Proximitätsmaße ermöglichen es, die Ähnlichkeit bzw. Unähnlichkeit von Objekten durch reelle Zahlen auszudrücken. Der Grad der Verschiedenheit oder Ähnlichkeit zweier Objekte A mit Merkmalsvektor $x = (x_1, \ldots, x_p)$ und B mit Merkmalsvektor $y = (y_1, \ldots, y_p)$ wird entweder durch ein Distanzmaß $d = (x, y)$ oder ein Ähnlichkeitsmaß $s = (x, y)$ wiedergegeben. Backhaus et al. liefern folgende Definitionen:

> „*Ähnlichkeitsmaße* spiegeln die Ähnlichkeit zwischen zwei Objekten wider: Je größer der Wert eines Ähnlichkeitsmaßes wird, desto ähnlicher sind sich zwei Objekte. *Distanzmaße* messen die Unähnlichkeit zwischen zwei Objekten: Je größer die Distanz wird, desto unähnlicher sind sich zwei Objekte." (Backhaus et al. 2003, 483)

$$\begin{pmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & 0 & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & 0 & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & d_{m3} & \cdots & 0 \end{pmatrix}$$

Abbildung 6.1: Proximitätsmatrix

Als Ergebnis einer Proximitätsberechnung erhält man eine symmetrische (da $s_{ij} = s_{ji}$ bzw. $d_{ij} = d_{ji}$) $m \times n$ Proximitätsmatrix, wobei m die Anzahl der Objekte und n die Anzahl der Attribute pro Objekt bezeichnet (siehe Abbildung 6.1). Solch eine Proximitätsmatrix findet z.B. bei hierarchischen Fusionierungsverfahren (vgl. Kapitel 7.1.1) Anwendung.

Hat man Ähnlichkeitswerte ermittelt und möchte diese in Distanzwerte umwandeln, bieten sich für $s_{xy} \in [0; 1]$ u.a. folgende Möglichkeiten der Transformation an (nach Panyr 1986, 56):

$$d_{xy} = 1 - s_{xy} \qquad \text{oder} \qquad d_{xy} = \sqrt{1 - s_{xy}}$$

Everitt et al. (2001, 43) weisen darauf hin, dass beispielsweise durch die zweite genannte Transformation aus einer bestimmten, nicht-negativen Ähnlichkeitsmatrix S eine Distanzmatrix D gewonnen werden kann, deren Distanzen metrisch sind. Ansonsten ist eine Transformation zu metrischen Distanzen nicht zwangsläufig gegeben.

Es wurde eine Vielzahl von Distanz- oder Ähnlichkeitsmaßen entwickelt, die für bestimmte Datentypen und Skalen geeignet sind. Im Folgenden wird eine Auswahl von Ansätzen zur Distanzberechnung zwischen Objekten vorgestellt.

6.1 Eigenschaften von Distanzmaßen

Findet die Distanzberechnung in einem metrischen Raum statt, muss ein Distanzmaß als Metrik nachfolgend aufgeführte Bedingungen erfüllen (vgl. Walz 2001, 419): Sei X eine beliebige Menge. Eine Abbildung $d : X \times X \rightarrow \mathbf{R}$ heißt Metrik, wenn für beliebige Elemente x, y und z von X gilt:

(i)	$d(x,y) \geq 0$	(Abstände können nicht negativ sein)
(ii)	aus $d(x,y) = 0$ folgt $x = y$	(Definitheit)
(iii)	$d(x,y) = d(y,x)$	(Symmetrie)
(iv)	$d(x,y) \leq d(x,z) + d(z,y)$	(Dreiecksungleichung)

Die Dreiecksungleichung besagt, dass der Abstand von X zu Y stets kleiner oder gleich dem Abstand über den „Umweg“ von Y nach Z und von Z nach Y ist (Abbildung 6.2). Anders ausgedrückt: Der direkte Weg ist immer der Kürzeste.

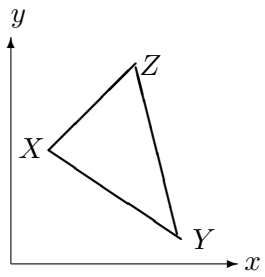


Abbildung 6.2: Dreiecksungleichung

Diese Eigenschaften scheinen insgesamt gesehen trivial zu sein. Sie gleichen dem Distanzbegriff unserer alltäglichen Erfahrungswelt, da wir Abstände in einem zwei- oder dreidimensionalen Raum mittels der Euklidischen Distanz berechnen, die sämtliche geforderten Eigenschaften aufweist. Verschärft man Bedingung (iv) zu $d(x,y) \leq \max\{d(x,z), d(y,z)\}$, erhält man eine Ultrametrik.

Im Weiteren werden neben den weit verbreiteten metrischen auch nicht-metrische Ansätze zur Distanzberechnung vorgestellt.

6.2 Minkowski-Metriken

Bei Objekten mit metrischem Skalenniveau wird am häufigsten als Proximitätsmaß eine Variante der Minkowski-Metrik oder L_p-Metrik eingesetzt. Generell ist sie definiert durch:

$$d_{ij} = \left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^p \right)^{1/p} \qquad (p \geq 1),$$

wobei x_{ik}, x_{jk} dem i-ten bzw. j-en Wert der k-ten Variablen entspricht und die Differenzen über die Gesamtzahl aller m Dimensionen aufsummiert werden.

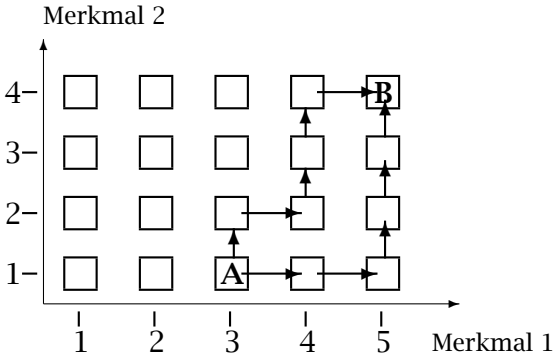


Abbildung 6.3: City-Block-Metrik (vgl. Deichsel und Trampisch 1980, 24)

Mit den Parametern $p = 1$ erhält man die L_1-Norm, die unter City-Block-Metrik, Manhattan-Metrik oder Taxifahrer-Metrik bekannt ist. Dabei wird die Distanz als kürzester Weg zwischen zwei Punkten (hier A(3,1) und B(5,4)) in einem zweidimensionalen Raum berechnet, ohne dass ein „Umweg“ gegangen wird[1] (Abbildung 6.3). Die Distanz nach der City-Block-Metrik errechnet sich wie folgt:

$$d(A,B) = \sum_{k=1}^{2} |x_{ik} - x_{jk}| = |3-5| + |1-4| = 2 + 3 = 5$$

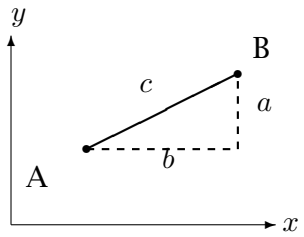


Abbildung 6.4: Euklidische Distanz

Für $p = 2$ erhält man die L_2-Norm oder die Euklidische Distanz, die im zweidimensionalen und dreidimensionalen Raum anschaulich der Berechnung der „LuftlinienEntfernung“ mit Hilfe des Satzes von Pythagoras gleicht (Abbildung 6.4). Für die Punkte A(3,1) und B(5,4) erhält man folgendes Ergebnis:

$$\begin{aligned} d(A,B) &= \sum_{k=1}^{2} \left((x_{ik} - x_{jk})^2 \right)^{1/2} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \\ &= \sqrt{(-2)^2 + (-3)^2} = \sqrt{13} \end{aligned}$$

Nach Backhaus et al. ist die mit der Euklidischen Distanz verbundene interne Quadrierung der Summanden von Vorteil, denn „Durch die Quadrierung werden große

[1] Eine andere Vorstellungshilfe ist, dass ein Taxifahrer, der in einer Stadt mit rechtwinklig zueinander laufenden Straßen von A nach B gelangen will, diese Entfernung zurücklegen muss (Bortz 1989).

Differenzwerte bei der Berechnung der Distanz stärker berücksichtigt, während geringen Distanzwerten ein kleineres Gewicht zukommt.“ (Backhaus et al. 2003, 493) Wird aus Gründen der rechnerischen Einfachheit mit der quadrierten Euklidischen Distanz gearbeitet, ist diese Distanz keine L_p-Metrik und keine metrische Distanz mehr (vgl. Kaufmann und Pape 1984, 384).

Bei Anwendung der Minkowski-Metriken muss bedacht werden, dass sie nicht skaleninvariant sind, d.h., das Ergebnis wird durch die Maßeinheit beeinflusst, in der die Merkmale gemessen werden (beispielsweise ob ein Merkmal in Zentimetern oder Metern erfasst wird). Ein Maß ist skaleninvariant, wenn sich die Distanzen monoton ändern und eine bestehende Ordnungsrelation erhalten bleibt. Beim Übergang von einem angelsächsischen zum metrischen Maßsystem ist dies z.B. nicht der Fall. Um die Ordnungsrelation beizubehalten, muss man entweder die Ausgangsdaten durch Standardisierung bzw. Normierung vergleichbar machen (vgl. Kapitel 5.4) oder ein skaleninvariantes Maß einsetzen, wie z.B. die Mahalanobis-Distanz (vgl. Deichsel und Trampisch 1980, 22).

6.3 Mahalanobis-Distanz

Die Mahalanobis-Distanz ist wie folgt definiert:

$$d_{ij} = (x_i - x_j)^T S^{-1} (x_i - x_j),$$

wobei S der empirischen Kovarianzmatrix

$$S = \frac{1}{N-1} \sum_{k=1}^{N} (x_i - \overline{x})(x_i - \overline{x})$$

entspricht und der Mittelwert $\overline{x}$ mittels $\overline{x} = \frac{1}{N} \sum_{k=1}^{N} x_i$ berechnet wird.

Eine vorteilhafte Eigenschaft dieser Distanz besteht darin, dass die bei der Berechnung einfließenden Merkmale unkorreliert sind, obwohl in den Ausgangsdaten eine Korrelation zwischen Merkmalen bestehen kann. Kaufmann und Pape begründen dies damit, dass die Ausgangsmerkmale zuerst transformiert werden und dadurch unkorrelierte Merkmale entstehen mit denen anschließend die quadrierte Euklidische Distanz berechnet wird (vgl. Kaufmann und Pape 1984, 385).

Hösel und Walcher weisen auf eine nachteilige Eigenschaft hin. Durch Anwendung einer gemeinsamen Kovarianzmatrix wird die Grundidee vom Vorhandensein verschiedener Klassen unterlaufen. Für jeden einzelnen Cluster müsste eine eigene Kovarianzmatrix erstellt werden, da ein Merkmal innerhalb einer Klasse unterschiedlich verteilt ist; dies stellt besondere Anforderungen an die Berechnungsalgorithmen (vgl. Hösel und Walcher, 5).

6.4 Ähnlichkeitsmaße bei binären Merkmalen

Binäre Merkmale treten in Zusammenhang mit nominal skalierten Merkmalen auf und geben mit ihren möglichen zwei Ausprägungen wieder, ob ein Merkmal vorhanden ist ($= 1$) oder nicht vorhanden ist ($= 0$). Um die Ähnlichkeit von zwei Objekten zu ermitteln, werden sämtliche Komponenten ihrer Merkmalsvektoren ($X = (x_{i1} \dots x_{in})$ und $Y = (y_{i1} \dots y_{in})$) miteinander verglichen. Es können dabei vier Fälle auftreten, die zur Übersicht in einer Kontingenztafel (Tabelle 6.1) zusammengefasst werden:

(a) Merkmal in beiden Objekten vorhanden
(b) Merkmal nur in Objekt 2 vorhanden
(c) Merkmal nur in Objekt 1 vorhanden
(d) Merkmal in beiden Objekten nicht vorhanden

	Objekt 2		Zeilensumme
Objekt 1	Eigenschaft vorhanden (1)	Eigenschaft nicht vorhanden (0)	
Eigenschaft vorhanden (1)	a	c	$a + c$
Eigenschaft nicht vorhanden (0)	b	d	$b + d$
Spaltensumme	$a + b$	$c + d$	m

Tabelle 6.1: Kontingenztafel für binäre Merkmale (Backhaus et al. 2003, 484)

Ähnlichkeitsmaße für binäre Merkmale sind beispielsweise der „simple matching"- oder M-Koeffizient

$$s_{ij} = \frac{a + d}{m},$$

bei dem das Vorhandensein und Nicht-Vorhandensein von Merkmalen gleichermaßen gewichtet wird. Ein weiteres Maß ist der Jaccard- bzw. S-Koeffizient

$$s_{ij} = \frac{a}{a + b + c},$$

der geeigneter ist, wenn eine Vielzahl von Dimensionen m vorliegt und das Ergebnis nicht durch das Mitzählen gemeinsam fehlender Merkmale beeinflusst werden soll (vgl. Hösel und Walcher, 6). Ein Beispiel zur angemessenen Verwendung führen Steinbach et al. an:

> „[...] if the vectors represent student's answers to a True-False test, then both 0-0 and 1-1 matches are important and these two students are very similar, at least in terms of the grades they will get. If instead the vectors indicate particular items purchased by two shoppers, then the Jaccard

> measure is more appropriate since it would be odd to say that the purchasing behavior of two customers is similar, even though they did not buy any of the same items.“ (Steinbach et al. 2002, 8)

Die in diesem Kapitel beschriebenen Ähnlichkeitsmaße finden kaum Anwendung bei der Verarbeitung von Dokumenten, beispielsweise im Information Retrieval. In diesem Bereich werden erfolgreich andere Arten der Ähnlichkeitsberechnung eingesetzt, auf die im folgenden Kapitel eingegangen wird.

6.5 Ähnlichkeitsmaße im Vektorraummodell

Im IR werden zu einer Anfrage als Ergebnis Dokumente geliefert. Dabei müssen Ähnlichkeiten zwischen der Anfrage und den gespeicherten Dokumenten in der Datenbasis berechnet werden. Beim Clustering werden die Dokumente nicht mit einer Anfrage verglichen, sondern jeweils die Dokumente bzw. die Cluster-Repräsentanten miteinander, um z.B. eine Proximitätsmatrix zu berechnen. Diese Proximitätsmatrix ist wiederum Ausgangspunkt für die hierarchischen Fusionierungsverfahren zur Gruppenbildung. Zur Repräsentation der Dokumente und Anfragen bzw. Cluster wird in beiden Fällen das Vektorraummodell (vgl. Kapitel 5.1) herangezogen, dessen Möglichkeiten zur Ähnlichkeitsberechnung von Objekten im Folgenden vorgestellt werden.

Die Maße können entweder mit binären Vektoren als Ausgangsdaten oder mit reellwertigen Vektoren berechnet werden. Für die im Rahmen dieser Arbeit durchgeführten Experimente (siehe Kapitel 8) werden die Terme der Patentdokumente, die die Datengrundlage darstellen, gewichtet. Aus diesem Grund sind die Dokumentenvektoren, die ein Patentdokument repräsentieren, mit reellen Zahlenwerten besetzt. Sollen binäre Vektoren eingesetzt werden, so wird das Vorkommen eines Terms mit dem Wert 1, das Nicht-Vorhandensein eines Terms mit dem Wert 0 gekennzeichnet. Einen Überblick über die Möglichkeiten zur Ähnlichkeitsberechnung im Vektorraummodell liefert Tabelle 6.2 (Variablen umbenannt, ansonsten zitiert nach Haenelt 2003). Für $|X \cap Y|$, die Schnittmenge zweier Merkmalsvektoren, kann man analog zur Darstellung der Kontingenztafel im vorangegangenen Kapitel die Benennung a verwenden.

Der Cosinus-Koeffizient ist ein Art Korrelationsmaß, das Ergebnisse im Intervall $[-1, 1]$ liefert. Ein Wert von 1 entspricht der größtmöglichen Ähnlichkeit zweier Objekte (Vektoren zeigen in die gleiche Richtung); nimmt der Cosinus-Koeffizient einen Wert von -1 an, so spiegelt dies die maximale Unähnlichkeit von zwei Objekten wider (Vektoren zeigen in entgegengesetzte Richtungen) (vgl. Everitt et al. 2001, 41).

Maß	binäre Vektoren	Vektoren mit reellen Werten
Dice-Koeffizient	$\frac{2\|X \cap Y\|}{\|X\|+\|Y\|}$	$\frac{2\sum_{i=1}^{n}(weight_{xi} \cdot weight_{yi})}{\sum_{i=1}^{n} weight_{xi}^2 + \sum_{i=1}^{n} weight_{yi}^2}$
Overlap-Koeffizient	$\frac{\|X \cap Y\|}{\min(\|X\|,\|Y\|)}$	$\frac{\sum_{i=1}^{n} \min(weight_{xi} \cdot weight_{yi})}{\min\left(\sum_{i=1}^{n} weight_{xi}, \sum_{i=1}^{n} weight_{yi}\right)}$
Cosinus-Koeffizient	$\frac{\|X \cap Y\|}{\sqrt{\|X\| \times \|Y\|}}$	$\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} x_i^2}}$

Tabelle 6.2: Ähnlichkeitsmaße im Vektorraummodell

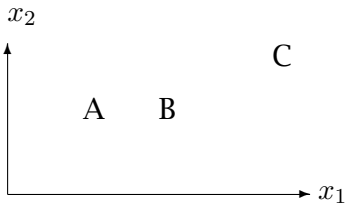


Abbildung 6.5: Mutual Neighbor Distance - A und B sind ähnlicher als A und C (Jain et al. 1999, 273)

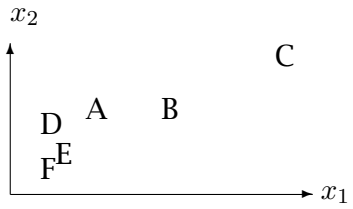


Abbildung 6.6: Mutual Neighbor Distance – Nach Veränderung des Kontexts: B und C sind ähnlicher als B und A. (Jain et al. 1999, 273)

6.6 Mutual Neighbor Distance-Verfahren

Eine weitere Möglichkeit der Distanzberechnung besteht darin, die Umgebung der Objekte (Kontext) mit einfließen zu lassen. Jain et al. stellen stellvertretend das *mutual neighbor distance*-Verfahren[2] (MND) vor, bei dem sich die Distanz der Objekte nach der Anzahl der Nachbarn in direkter Umgebung berechnet:

$$MND(x_i, x_j) = NN(x_i, x_j) + NN(x_i, x_j),$$

wobei $NN(x_i, x_j)$ der Anzahl nächster Nachbarn von Objekt x_j hinsichtlich Objekt x_j entspricht. Der erfolgreiche Einsatz in Anwendungen legt den Schluss nahe, dass nicht zwangsläufig ein metrisches Maß zur Distanzbestimmung vorliegen muss. In Abbildung 6.5 gilt: $MND(A, B) = 2$, da $NN(A, B) = NN(B, A) = 1$ und $MND(B, C) = 3$, da $NN(B, C) = 1$ und $NN(C, B) = 2$. Nach Hinzufügen der Punkte D, E, und F errechnet sich der Abstand neu (Abbildung 6.6) zu $MND(B, C) = 5$ und $MND(A, B) = 5$ (vgl. Jain et al. 1999, 273).

[2] erstmals beschrieben in: Gowda, K. C., Krishna, G. (1978): Agglomerative Clustering Using the Concept of Mutual Nearest Neighborhood. Pattern Recognition. Vol. 10, pp. 105-112.

6.7 Weitere Proximitätsmaße

Weitere Verfahren zur Distanzbestimmung bieten die Canberra-Metrik oder der Pearson-Korrelationskoeffizient bzw. Q-Korrelationskoeffizient (vgl. Everitt et al. 2001, 41). Everitt et al. bezeichnen die Canberra-Metrik, die durch

$$d_{ij} = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|} \qquad x_{ik} \neq 0 \quad \text{oder} \quad x_{jk} \neq 0$$

definiert ist, als ein Maß, das sehr empfindlich auf kleine Veränderungen im Umfeld von $x_{ik} = x_{jk} = 0$ reagiert oder oft als generalisiertes Distanzmaß für binäre Merkmale verwendet wird. Der Pearson-Korrelationskoeffizient, der mittels

$$\varphi_{ij} = \frac{\sum_{k=1}^{p}(x_{ik} - \overline{x}_i)(x_{jk} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{p}(x_{ik} - \overline{x}_i)^2 \sum_{k=1}^{p}(x_{jk} - \overline{x}_j)^2}} \qquad \text{wobei} \quad \overline{x}_i = \frac{1}{p}\sum_{k=1}^{p} x_{ik}$$

berechnet wird, liefert durch Umformung mittels $d_{ij} = (1 - \varphi_{ij})/2$ einen Distanzwert. Der Unterschied zwischen diesem und dem Cosinus-Koeffizienten liegt darin, dass vom Mittelwert der Vektoren (d.h. dem Durchschnittswert aller Eigenschaften bei Objekt i bzw. j) ausgegangen wird, und nicht wie beim Cosinus-Koeffizienten vom Ursprung der zu vergleichenden Vektoren.

6.8 Zusammenfassung

Zur Bestimmung der Ähnlichkeit bzw. Unähnlichkeit von Datenobjekten können die in diesem Kapitel vorgestellten Proximitätsmaße eingesetzt werden. Bei der Auswahl eines Maßes ist der Skalentyp der Attribute (vgl. Kapitel 5.2) zu berücksichtigen, denn nicht alle Proximitätsmaße sind für jeden Skalentyp geeignet (z.B. wenn binäre Merkmale vorliegen). Für das Clustern von Dokumenten wird überwiegend der Kosinus-Koeffizient als Ähnlichkeitsmaß herangezogen, so auch für ein Clustering-Verfahren, das im Rahmen dieser Arbeit getestet wurde (vgl. Kapitel 8).

Im nächsten Kapitel werden Verfahren zur Gruppenbildung, d.h. zum Erzeugen von Clustern, vorgestellt, die hierfür auf die berechneten Ähnlichkeiten bzw. Unähnlichkeiten zurückgreifen.

7 Fusionierungsverfahren

Die ermittelte Distanz- bzw. Ähnlichkeitsmatrix stellt den Ausgangspunkt für viele Clustering-Verfahren dar, die die Objekte in Gruppen zusammenfassen (Backhaus et al. 2003, 499). Einige dieser Fusionierungsalgorithmen arbeiten auf einer Proximitätsmatrix (so z.B. hierarchische Verfahren), wohingegen andere Algorithmen wiederum andere Ausgangspunkte für das Bilden von Gruppen wählen. Eine Einteilung der verschiedenen Algorithmen hinsichtlich der zu Grunde liegenden Fusionierungsverfahren nehmen Backhaus et al. (2003, 499) vor (Abbildung 7):

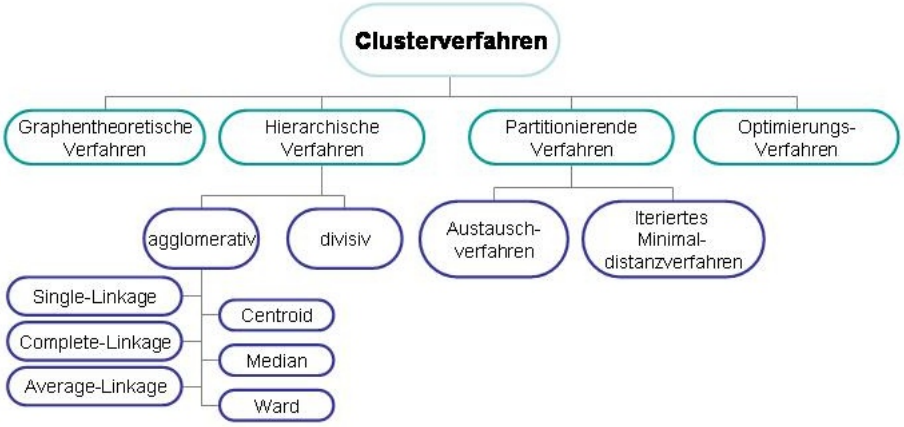


Abbildung 7.1: Überblick über ausgewählte Clustering-Algorithmen (Backhaus et al. 2003, 499)

Jain et al. (1999, 274) beschreiben Merkmale, nach denen Clustering-Algorithmen weiter unterteilt werden können:

- Werden beim Fusionierungsprozess – wie bei den meisten Algorithmen anzutreffen – sämtliche Variablen gleichzeitig mit einbezogen, so spricht man von *polythetischen* Verfahren. Wird hingegen sequentiell jeweils nur eine Variable herangezogen, so spricht man von *monothetischen* Verfahren. Für hochdimensionale Daten, wie sie z.B. im Information Retrieval anfallen, sind monothetische Verfahren ungeeignet, da sie zu kleine und fragmentierte Cluster erzeugen (vgl. Jain et al. 1999, 274). Der Vorteil der Clusteranalyse, so Backhaus et al. (2003, 499), ist gerade die simultane Betrachtung aller Merkmale, weshalb sich diese Autoren in ihren Ausführungen ausschließlich auf polythetische Verfahren konzentrieren.
- Ein Objekt kann entweder einem Cluster fest zugeteilt werden *(hard clustering)* oder für jede Instanz wird eine graduelle Zugehörigkeit zu einem oder mehreren Clustern berechnet *(fuzzy clustering)* (vgl. Kapitel 7.5.1).

- Als viele Clustering-Algorithmen entwickelt wurden, gab es noch nicht die heutzutage sehr häufig anzutreffenden riesigen Datenmengen. Zunächst wurden daher *inkrementelle* Verfahren formuliert, die jedoch mit den wachsenden Datenmengen nicht mehr Schritt halten können, weil z.B. mehrere Durchläufe über sämtliche Eingangsdaten zum Berechnen der Distanzen benötigt wurden. Besonders bei einer großen Anzahl von Daten ist eine effiziente Verarbeitung wichtig, die von *nicht-inkrementellen* Verfahren beispielsweise in Form kleinerer Datenstrukturen und einer Reduktion der Durchläufe über die Eingangsdaten erreicht wird.

7.1 Hierarchische Verfahren

Die Gruppe der hierarchischen Fusionierungsverfahren wird in diesem Kapitel vorgestellt. Dazu werden zunächst Grundlagen und Eigenschaften dieser Verfahrensgruppe beschrieben, um anschließend die unterschiedlichen Möglichkeiten zur Berechnung der Distanzen bzw. Ähnlichkeiten zwischen einzelnen Clustern aufzuzeigen.

7.1.1 Grundlagen hierarchischer Verfahren

Hierarchische Clustering-Algorithmen liefern als Ergebnis ineinander verschachtelte Cluster, die graphisch durch ein Dendrogramm dargestellt werden (siehe Abbildung 7.2). Ein Dendrogramm gibt diese hierarchische Anordnung in Form eines umgedrehten Baumes wieder (Wurzel liegt oben) und veranschaulicht, an welchen Knotenpunkten Cluster zusammengeführt bzw. geteilt werden. Die Höhe im Dendrogramm kann als Homo- bzw. Heterogenitätsmaß betrachtet werden: Je nachdem, in welcher Höhe ein Schnitt im Dendrogramm erfolgt, erhält man eine unterschiedliche Anzahl an Clustern, die auf dieser Stufe ein bestimmtes Maß an Homogenität aufweisen (gestrichelte Linie in Abbildung 7.2). Es ist daher nicht notwendig, die Anzahl der zu erzeugenden Cluster im Vorfeld anzugeben, wie dies bei den partitionierenden Verfahren (siehe Kapitel 7.2) erforderlich ist. Dies mag als Vorteil für hierarchische Verfahren gewertet werden. Jedoch muss man sich entscheiden, in welcher Höhe ein Schnitt[1] im Dendrogramm erfolgen soll, um die „geeignetste“ Clusteranzahl zu erhalten.

Zur Erzeugung einer Hierarchie gibt es zwei Ansätze:

1. **agglomeratives Verfahren:** Zu Beginn repräsentiert jedes Objekt einen eigenen Cluster. Nach und nach werden ähnliche Objekte zu größeren Clustern

[1] Für eine formale Definition einer Hierarchie und des Schnittes auf einer Stufe im Dendrogramm, vgl. Panyr (1986, 80 f.).

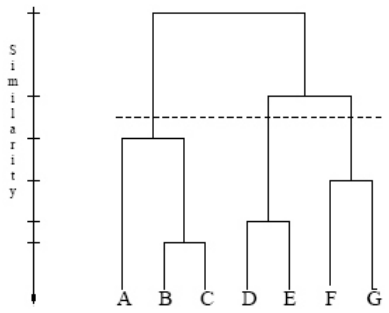


Abbildung 7.2: Dendrogramm (Jain et al. 1999, 276)

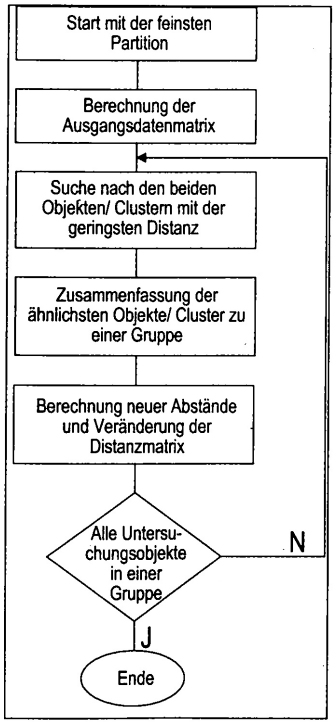


Abbildung 7.3: Ablauf des hierarchisch-agglomerativen Clustering-Verfahrens (Backhaus et al. 2003, 504)

verschmolzen, bis als Endpunkt sämtliche Objekte in einem großen Cluster zusammengeführt sind *(bottom-up Ansatz)*. Der generelle Ablauf eines hierarchisch-agglomerativen Verfahrens wird zur Verdeutlichung als Struktogramm in Abbildung 7.3 visualisiert.

2. **divisives Verfahren:** Alle Objekte befinden sich zu Beginn in einem großen, allumfassenden Cluster. Ein großer Cluster wird solange in kleinere Cluster aufgeteilt, bis sämtliche Objekte in einem eigenen Cluster liegen *(top-down Ansatz)*. Divisive Verfahren sind auf Grund der Vielzahl von Aufteilungsschritten aufwändiger zu berechnen und werden hauptsächlich bei binären Merkmalen eingesetzt (vgl. Hösel und Walcher, 10).

Ein großer Nachteil von hierarchischen Verfahren liegt darin, dass einmal getroffene Entscheidungen über das Aufspalten oder Verschmelzen von Clustern nicht mehr

rückgängig gemacht werden können.[2] Im Gegensatz dazu stehen die partitionierenden Verfahren, bei denen ein Objekt jederzeit zwischen den Clustern verschoben werden kann, falls dies zu einem besseren Gesamtergebnis führt. Für hierarchische Verfahren spricht wiederum, dass sie im Vergleich zu den partitionierenden Verfahren nicht in lokalen Minima stecken bleiben können oder dass das Endergebnis von der Auswahl der Initialpunkte unabhängig ist (vgl. Kumar 2003, 321 f.).

Die in den folgenden Abschnitten vorgestellten *Methoden und Verfahren* zum Clustern von Daten können durch unterschiedliche *Algorithmen* implementiert werden (vgl. Abschnitt 16.5 Rasmussen 1992). Generell werden hierarchische Verfahren als sehr anspruchsvoll in Bezug auf ihre Speicherplatzanforderungen und ihr Laufzeitverhalten beschrieben (vgl. Jain et al. 1999, 293). Der Speicherplatzbedarf zur Berechnung der Distanzmatrix wird je nach Algorithmus (in Abhängigkeit zur Anzahl der Eingabedaten N bzw. n) zwischen $O(N)$ und $O(N^2)$ angegeben (vgl. Rasmussen 1992). Der Zeitbedarf schwankt zwischen $O(N^2 \log n)$ im „best-case“ und $O(N^3)$ im „worst-case“ (vgl. Jain et al. 1999, 293).[3]

7.1.2 Verfahren zur Bestimmung der inter-Cluster Proximität

Die hierarchisch-agglomerativen Clustering-Verfahren unterscheiden sich in der Art und Weise, wie aus den intra-Cluster Ähnlichkeits- bzw. Distanzwerten (z.B. in Form einer Distanzmatrix errechnet mit Verfahren aus Kapitel 6) die Klassen gebildet werden. Unterschiedliche Linkage-Algorithmen werden in den folgenden Abschnitten vorgestellt auf deren Basis Objekte zu Clustern bzw. Cluster miteinander verschmolzen werden.

Die gängigen agglomerativen Linkage-Algorithmen lassen sich rekursiv mittels der von Lance und Williams aufgestellten Formel mit entsprechender Wahl der Parameter (Lance-Williams Koeffizienten siehe Tabelle 7.1) formulieren.

Die Unähnlichkeit zwischen einer durch Verschmelzen entstandenen Klasse C ($C = C_i \cup C_j$) und einer weiteren Klasse C_s wird folgendermaßen ermittelt (Kaufmann und Pape 1984, 393):

$$d_{C_i,C_s} = \alpha_i d_{C_i,C_s} + \alpha_j d_{C_j,C_s} + \beta d_{C_i,C_j} + \gamma |d_{C_i,C_s} - d_{C_j,C_s}|$$

[2] „A hierarchical method suffers from the defect that it can never repair what was done in previous steps.“ Kaufman/Rousseeuw, zitiert nach Everitt et al. (2001, 55).
[3] Zur Definition der $O(x)$-Notation (vgl. Day 1996, 207 ff.).

Methode	α_i	α_j	β	γ
Single L.	$1/2$	$1/2$	0	$-1/2$
Complete L.	$1/2$	$1/2$	0	$1/2$
Average L.	$n_j/(n_i+n_j)$	$n_j/(n_i+n_j)$	0	0
Centroid	$n_j/(n_i+n_j)$	$n_j/(n_i+n_j)$	$-n_in_j/(n_i+n_j)^2$	0
Median	$1/2$	$1/2$	$-1/4$	0
Ward	$(n_i+n_s)/(n_i+n_j+n_s)$	$(n_j+n_s)/(n_i+n_j+n_s)$	$-(n_s)/(n_i+n_j+n_s)$	0

Tabelle 7.1: Parameter der Lance-Williams Formel für hierarchisch agglomerative Clustering-Verfahren (Kaufmann und Pape 1984, 394)

7.1.2.1 Single Linkage-Verfahren

Beim *Single Linkage-Verfahren*[4] (alternativ: nearest neighbor method, minimum distance method), siehe Abbildung 7.4, „ist die Distanz zwischen den Klassen C_k und C_j gleich der kleinsten Distanz zwischen einem Objekt aus C_k und einem Objekt aus C_j:“ (Kaufmann und Pape 1984, 395)

$$d(C_k, C_j) = \min_{n \in C_k, m \in C_j} d_{nm}$$

Abbildung 7.4: Single Linkage

Eigenschaften des Single Linkage-Algorithmus:

- Cluster beliebiger Form werden erkannt (vgl. Kaufmann und Pape 1984, 396). Deswegen bietet Single Linkage eine größere Flexibilität, da z.B. auch konzentrisch angeordnete Clusterstrukturen ermittelt werden können, was bei anderen Linkage-Verfahren nicht möglich ist (vgl. Abbildung 7.5) (vgl. Jain et al. 1999, 276).
- Liegen im Raum zwischen den Klassen einige wenige Objekte, kann der Single Linkage-Algorithmus die dazwischenliegenden Objekte als „Brücke“ interpretieren und so zu einer heterogeneren Klasseneinteilung gelangen, obwohl eine eindeutig homogenere Aufteilung in Klassen möglich gewesen wäre. In Abbildung 7.6 soll dieser Effekt durch die verrauschten Daten (*) als Brücke veranschaulicht werden. Man bezeichnet dies als „Chaining-Effekt“ (vgl. Jain et al. 1999, 276).
- „It has a tendency to produce clusters that are straggly or elongated.“ (Jain et al. 1999, 276) Backhaus et al. (2003) sehen in dieser Eigenschaft eine zusätzliche Anwendungsmöglichkeit: „Da das Single-Linkage-Verfahren dazu neigt, viele kleine und wenige große Gruppen zu bilden (*kontrahierendes* Verfahren), bilden die kleinen Gruppen einen Anhaltspunkt für die Identifikation von ‚Ausreißern’ in der Objektmenge.“ (Backhaus et al. 2003, 509). Nachdem die „Ausreißer“ eliminiert wurden, kann z.B. durch ein Complete Linkage-Verfahren eine bessere Klassenaufteilung erreicht werden.

[4] zuerst beschrieben von Sneath in: Sneath, P.H.A. (1957): The application of computers to taxonomy. Journal of General Microbiology, 17, 2001-226. (nach Everitt et al. 2001, 62)

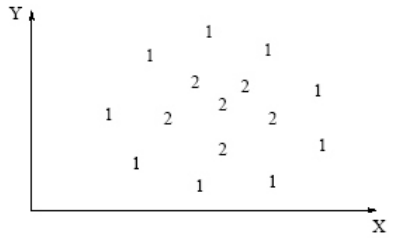


Abbildung 7.5: konzentrisch angeordnete Cluster (Jain et al. 1999, 276)

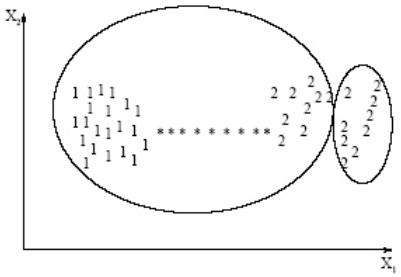


Abbildung 7.6: Ergebnis, das mit dem Single Linkage-Verfahren entsteht. Zwei Klassen (1 und 2) sind über eine Brücke von Rauschen (*) verbunden (Jain et al. 1999, 277).

7.1.2.2 Complete Linkage-Verfahren

Beim *Complete Linkage-Verfahren*[5] (alternativ: furthest neighbor method, maximum distance method, siehe Abbildung 7.7, „ist die Distanz zwischen den Klassen C_k und C_j gleich der größten Distanz zwischen einem Objekt aus C_k und einem Objekt aus C_j:“ (Kaufmann und Pape 1984, 396)

$$d(C_k, C_j) = \max_{n \in C_k, m \in C_j} d_{nm}$$

Abbildung 7.7: Complete Linkage

Eigenschaften des Complete Linkage-Algorithmus:

- Im Gegensatz zum Single Linkage Algorithmus ermittelt das Complete LinkageVerfahren eher kleinere Gruppen, die ungefähr gleich groß sind („[It] Tends to find compact clusters with equal diameters (maximum distance) between objects.“ (Everitt et al. 2001, 62)). Backhaus et al. (2003) bezeichnen solche Verfahren als *dilatierend*.
- Dem Ergebnis von Complete Linkage-Verfahren wird eine bessere Qualität zugeschrieben: „[. . .] from a pragmatic viewpoint, it has been observed that the complete-link algorithm produces more useful hierarchies in many applications than the single-link algorithm.“ (Jain et al. 1999, 276)
- Es tritt kein Chaining-Effekt auf, wie es beim Single Linkage-Verfahren zu beobachten ist (Abbildung 7.8).

[5] erstmals beschrieben von Sorensen in: Sorensen, T. (1948): A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biologiske Skrifter, 5, 1-35. (nach Everitt et al. 2001, 62)

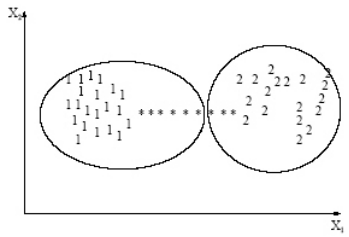


Abbildung 7.8: Ergebnis, das mit dem Complete-Linkage Verfahren entsteht. Zwei Klassen (1 und 2) sind über eine Brücke von Rauschen (*) verbunden (Jain et al. 1999, 277).

7.1.2.3 Average Linkage-Verfahren

Beim *Average Linkage-Verfahren*[6] (siehe Abbildung 7.9), auch bekannt als *unweighted pair-group method using the average approach (UPGMA)* (Everitt et al. 2001, 60), „ist die Distanz zwischen den Klassen C_k und C_j gleich dem Durchschnitt aller Distanzen zwischen Objekten aus C_k und C_j:“ (Kaufmann und Pape 1984, 397)

$$d(C_k, C_j) = \frac{1}{n_k n_j} \sum_{n \in C_k} \sum_{m \in C_j} d_{nm}$$

Abbildung 7.9: Average Linkage

Eigenschaften des Average Linkage-Verfahrens (vgl. Everitt et al. 2001, 62):

- „Tends to join clusters with small variances.
- Intermediate between single and complete linkage.
- Relatively robust.“

7.1.2.4 Centroid-Verfahren

Das *Centroid-Verfahren*[7] basiert darauf, dass jede Klasse durch ihren Centroid (= Klassenschwerpunkt) repräsentiert wird, der sich wie folgt errechnet: $\overline{x}_k = \frac{1}{n_k} \sum_{i \in C_k} x_n$ Es werden die Klassen miteinander verschmolzen, deren Centroide den geringsten Abstand aufweisen. Everitt et al. (2001) bezeichnen diese Verfahren auch als *unweighted*[8] *pair-group method using the centroid approach (UPGMC)* (vgl. Everitt et al. 2001, 60). Formal (vgl. Kaufmann und Pape 1984, 398):

$$d(C_k, C_j) = \min_{\overline{x}_k, \overline{x}_j}$$

[6] zuerst beschrieben von Sokal und Michener in: Sokal, R.R., Michener, C.D. (1958): A statistical method for evaluating systematic relationships. University of Kansas Science Bulletin, 28, 1409-1438. (nach Everitt et al. 2001, 62)

[7] zuerst beschrieben von Sokal and Michener in: a.a.O. (nach Everitt et al. 2001, 62)

[8] Die Unterscheidung zwischen „(un)weighted“ führte zu einer beträchtlichen terminologischen Verwirrung (siehe hierzu Vogel 1975, 305). Das „(un)weighted“ bezieht sich auf eine eventuelle Gewichtung der Mittelwerte bzw. Medianwerte.

7.1.2.5 Median-Verfahren

Werden statt eines Centroids die Median-Werte zur Repräsentation von Klassen eingesetzt, erhält man das *Median-Verfahren*[9], das von Everitt als *weighted*[10] *pair-group method using the centroid approach (WPGMC)* bezeichnet wird. Im Unterschied zum Centroid-Verfahren werden hierbei die Klassen-Repräsentanten zusätzlich gewichtet, was verhindern soll, dass Cluster mit einer großen Anzahl an Objekten über kleinere Cluster dominieren (vgl. Everitt et al. 2001, 60).

7.1.2.6 Verfahren von Ward

Beim *Verfahren von Ward*[11] wird zunächst für jede Klasse die Homogenität mittels eines Streuungsmaßes ermittelt. Im nächsten Schritt werden die Klassen fusioniert, die nach dem Verschmelzen den geringsten Verlust an Homogenität aufweisen, um heterogene Cluster zu verhindern. Im Nachfolgenden Schreibweise nach Everitt et al. (2001, 60 f.): Als „Gütefunktion“ soll der Zuwachs aller Fehlerquadratsummen

$$E = \sum_{m=1}^{g} E_m$$

minimiert werden. Die Fehlerquadratsummen werden für jeden Cluster mittels Berechnung der quadrierten euklidischen Distanz zwischen den Objekten eines Clusters zu dessen Centroid ermittelt:

$$E_m = \sum_{l=1}^{n_m} \sum_{k=1}^{p} ((x_{ml,k} - \overline{x}_{m,k})^2$$

Dieses Vorgehen entspricht dem K-Means Verfahren (siehe Kapitel 7.2.2) und ermöglicht eine „globale“ Sichtweise auf die Daten. Eigenschaften des Verfahrens von Ward:

- „Tests have shown it to be good at recovering cluster structure, though it is sensitive to outliers and poor at recovering elongated clusters.“ (Rasmussen 1992)
- „Ward’s technique tends to result in clusters of similar size. It is not well suited to find clusters with a small number of objects or clusters which are stretched in one direction.“ (Hösel und Walcher, 13)

[9] zuerst beschrieben von Gower: Gower, J.C. (1967): A comparison of some methods of cluster analysis. Biometrics, 23, 623-628. (nach Everitt et al. 2001, 62)
[10] vgl. Fußnote 8
[11] Ward, J.H. (1963): Hierarchical groupings to optimize an objective function. Journal of the American Statistical Association, 58, 236-244. (nach Everitt et al. 2001, 62)

7.2 Partitionierende Verfahren

Bei den partitionierenden Verfahren wird von einer Anfangspartition ausgegangen, die schrittweise verbessert wird, bis ein (lokales) Optimum erreicht wird. Die Ausgangspartition ist entweder aus einer zufälligen Auswahl der Objekte heraus oder als Ergebnis eines anderen Verfahrens (z.B. eines hierarchischen Verfahrens) entstanden. Diese Initial-Objekte stellen die ersten Punkte eines Clusters dar, wobei die Anzahl der Initial-Objekte der Anzahl der gewünschten Cluster entspricht.

Als Ergebnis erhält man keine ineinander verschachtelte Struktur, wie bei den hierarchischen Verfahren, sondern eine Aufteilung aller Objekte in m zuvor festgelegte Cluster oder Partitionen. Steinbach et al. (2000) weisen darauf hin, dass partitionierende Verfahren genauso dazu benützt werden können, Hierarchien zu erstellen und umgekehrt:

> „Of course, a hierarchical approach can be used to generate a flat partition of K clusters, and likewise, the repeated application of a partitional scheme can provide a hierarchical clustering. The bisecting K-means algorithm [...] is such an approach." (Steinbach et al. 2000, 4)

Partitionierende Verfahren weisen ein günstiges Laufzeitverhalten auf ($O(nm)$, wobei $n =$ Anzahl der Objekte und $m =$ Anzahl der Cluster) (Rasmussen 1992). Aus diesem Grund sind sie für eine effiziente Verarbeitung von großen Datenmengen geeignet (vgl. Jain et al. 1999, 278).

Eine exakte Bestimmung der optimalen Partition ist nicht möglich, da sonst alle erdenklichen Partitionen durchprobiert werden müssten, was einer vollständigen Enumeration aller Lösungen entspräche. Bereits für wenige Objekte N, die in g Cluster eingeteilt werden sollen, wächst die Zahl der möglichen Partitionen sehr schnell. Die Anzahl der möglichen Partitionen lässt sich durch

$$\frac{1}{g!}\sum_{k=0}^{g}(-1)^k\binom{g}{k}(g-k)^N \qquad (1 \leq g \leq N)$$

berechnen (STIRLINGSCHE Zahl zweiter Art), wobei zur Verdeutlichung des Wachstumsverhaltens ein paar Beispielwerte in Tabelle 7.2 aufgeführt sind (nach Kaufmann und Pape 1984, 405).

g / N	3	5	10
10	9330	179487	1
20	580.606.446	$4,306 * 10^{12}$	$5,918 * 10^{12}$
50	$1,196 * 10^{23}$	$7,401 * 10^{32}$	$2,616 * 10^{43}$
100	$8,590 * 10^{46}$	$2,316 * 10^{69}$	$2,756 * 10^{93}$

Tabelle 7.2: Anzahl der möglichen Partitionen von N Objekten in g Klassen

Die im Folgenden vorgestellten Verfahren stellen somit Heuristiken dar, die eine Verbesserung der Ausgangspartition dadurch erreichen, dass entweder ein globales Gütekriterium bzw. eine Zielfunktion (bezüglich der gesamten Clustereinteilung) oder nur eines Teilaspekts (z.B. durch Maximierung der Clusterhomogenität) berücksichtigt wird (lokales Gütekriterium) (vgl. Panyr 1986, 70). Ein Objekt kann – anders als bei den hierarchischen Verfahren (siehe Kapitel 7.1.1) – während des Fusionierungsprozesses seine Clusterzugehörigkeit (mehrfach) ändern. Die Zahl der Cluster, die ein Verfahren erzeugen soll, muss dem jeweiligen Verfahren als Parameter angegeben werden. Um Partitionen ausgehend von einer Anfangspartition zu erzeugen, bieten sich zwei Wege an:

Bei den **Austauschverfahren** oder „hill climbing Verfahren“ wird für jedes Objekt untersucht, ob durch Verschiebung in einen anderen Cluster ein Gütekriterium verbessert wird. Wenn ja, dann wird dieses Objekt in den betreffenden Cluster verschoben (= Austausch). Das Vergleichen und Austauschen wird solange wiederholt, bis keine Verbesserung mehr eintritt (siehe Algorithmus 1).

> „Die Austauschverfahren arbeiten relativ langsam, da in jedem Schritt (zu einer weiteren verbesserten Partition) lediglich ein einziges Objekt überführt werden kann.“ (Panyr 1986, 70)

Algorithmus 1: Austauschverfahren (in allgemeiner Formulierung nach Steinhausen und Langer 1977, 128)

Anfangspartition vorgeben.
Berechne Gruppenzentren.
wiederhole
  Prüfe für jedes Element, ob sich die Zielfunktion dadurch verbessern lässt, daß es in eine andere Gruppe verschoben wird. Wenn ja, so verschiebe es in die Gruppe mit der größten Verbesserung und berechne für die so entstandene Gruppierung die Gruppenzentren neu.
bis *n mal hintereinander kein Gruppenwechsel erfolgt ist.*

Sollen mehrere Objekte gleichzeitig umgruppiert werden, kann das **iterative Minimaldistanzverfahren** angewandt werden. Im Unterschied zu den Austauschverfahren werden erst zum Schluss die Clusterzentren neu berechnet, nicht nach jeder Änderung. In Algorithmus 2 werden die Schritte zur Erzeugung einer vorgegebenen Clusteranzahl m erläutert (nach Panyr 1986, 70):

Algorithmus 2: iterative Minimaldistanzverfahren (Panyr 1986, 70)

Zu einer vorgegebenen Anfangspartition Z^0 werden zunächst die zugehörigen Clusterzentren gebildet.
wiederhole
  Jedes Objekt O_i wird jenem Zentrum zugeordnet, das am nächsten bei O_i liegt. Die Clusterzentren werden neu errechnet.
bis *keine Veränderung der Gütefunktion mehr auftritt*

Ein Hauptproblem der partitionierenden Algorithmen besteht darin, dass je nach gewählter Anfangspartition ein anderes Endergebnis entstehen kann. Wählt man wie in Abbildung 7.10 die Objekte A, B und C als Anfangspartition, so erhält man als Ergebnis folgende Partitionen: {{A}, {B, C}, {D, E, F, G}} (gekennzeichnet durch die Ellipsen). Die optimalen Partitionen (gekennzeichnet durch ein Rechteck) mit {{A, B, C}, {D, E}, {F, G}} hätten durch Auswahl der Objekte A, D, F als Startkonfiguration ermittelt werden können (vgl. Jain et al. 1999, 278 f.). Ein weiteres Problem von partitionierenden Verfahren besteht darin, dass das Endergebnis von der Reihenfolge der Eingabedaten abhängt:

> „Man kann sich jedoch leicht vorstellen, daß dieser Einfluß besonders von den Elementen ausgeübt wird, die nicht deutlich zu clustern sind, da sie entweder weit außerhalb eines jeden Clusters liegen und damit den Schwerpunkt in ihrem jeweiligen Cluster stark beeinflussen oder weil sie zwischen zwei Clustern liegen.“ (Steinhausen und Langer 1977, 117)

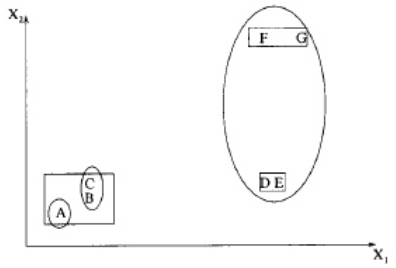


Abbildung 7.10: Abhängigkeit des K-Means Algorithmus von der Anfangspartition (Jain et al. 1999, 279)

Das Festlegen der Clusterzahl im Vorfeld und die Abhängigkeit des Ergebnisses von der Startpartition und Eingabereihenfolge der Daten stellen die methodischen Problempunkte der partitionierenden Verfahren dar. Um ein näherungsweise optimales Ergebnis zu erhalten, schlagen Jain et al. vor:

> „In practice, therefore, the algorithm is typically run multiple times with different starting states, and the best configuration obtained from all of the runs is used as the output clustering.“ (Jain et al. 1999, 278)

7.2.1 Gütefunktionen und Refinement-Phase

Zhao und Karypis (2001, 5 f.) untersuchten sechs Gütefunktionen bzw. Gütekriterien (engl. criterion function), die jeweils in drei Algorithmen eingesetzt wurden, anhand von fünfzehn verschiedenen Datensätze auf ihre Eignung zum Clustern von Dokumenten. Dabei unterschieden sie zwischen *internen*, *externen* und *hybriden* Gütekriterien, die alle im Programmpaket CLUTO (vgl. Kapitel A.1) realisiert sind.

Interne Gütekriterien versuchen, eine Funktion zu optimieren, „that is defined over the documents that are part of each cluster and does not take into account the documents assigned to different clusters." *Externe* Gütekriterien steuern die Erzeugung von Cluster-Lösungen, indem sie eine starke Unähnlichkeit zwischen den einzelnen Clustern positiv bewerten (indem z.B. die Centroide der Cluster sich vom Centroid der zu Grunde liegenden Dokumentenkollektion maximal unterscheiden sollen). *Hybride* Gütekriterien kombinieren die Ansätze von internen und externen Gütefunktionen.

Als Beispiel für eine interne Gütefunktion kann man das in Kapitel 6.5 eingeführte Cosinus-Maß anführen. Es berechnet die Ähnlichkeit eines Dokuments (d) und dem Centroid (C) eines Clusters, wobei diese Ähnlichkeitsbeziehung über alle Dokumente einer Kollektion (S) maximiert werden soll. Formal:

$$\mathcal{I}_2 = \sum_{r=1}^{k} \sum_{d_i \in S_r} \cos(d_i, C_r) \longrightarrow \max.$$

Traditionell wird in der Literatur zur Multivariaten Analyse als Gütefunktion das Varianzkriterium angeführt, das in der englischsprachigen Literatur als „squared-error criterion" bezeichnet wird und dem die Vorstellung zu Grunde liegt, „daß eine Klasse ähnlicher Objekte eine kleine Streuung innerhalb der Klasse aufweist." (Kaufmann und Pape 1984, 408). Dazu wird die euklidische Distanz zwischen einem Dokument (d) und dem Cluster-Centroid (C) berechnet, wobei diese Distanz für alle Dokumente einer Kollektion (S) minimal sein soll (vgl. Zhao und Karypis 2001, 5). Formal:

$$\mathcal{I}_3 = \sum_{r=1}^{k} \sum_{d_i \in S_r} |d_i - C_r|^2 \longrightarrow \min.$$

Als Ergebnis der Untersuchung durch Zhao und Karypis zeigt sich, dass die Gütefunktionen $\mathcal{I}_2$ und (die hier nicht aufgeführte) Gütefunktion $\mathcal{H}_2$ durchweg die besten Ergebnisse liefern, wohingegen die anderen Funktionen (auch das klassische Varianzkriterium $\mathcal{I}_3$) schlechte Ergebnisse erzielen.

Zur Verbesserung der Gütefunktion wählten Zhao und Karypis (2001, 8 f.) eine Optimierungsstrategie, die sie als **Refinement** bezeichneten. Ausgehend von der initialen Cluster-Lösung, bei der zur Erzeugung von k Clustern aus der Gesamtheit der Dokumente k Initial-Dokumente als Clusterrepräsentanten ausgewählt werden, wird nach jedem Durchlauf des partitionierenden Algorithmus eine Refinement-Phase angeschlossen. Dabei wird zufällig ein Dokument ausgewählt und überprüft, wie sich die Gütefunktion verändert, wenn dieses Dokument zu einem anderen Cluster gehören würde. Es wird letztlich in den Cluster verschoben, der die Gütefunktion am meisten verbessert. Die gesamte Operation (Ermitteln der Initiallösung und Refinement-

Phase) wird n Mal wiederholt. Die Lösung, die die Gütefunktion am meisten verbessert, dient in der nächsten Iteration als Ausgangspunkt des partitionierenden Verfahrens.

7.2.2 K-Means – eine auf Centroiden basierende Technik

Aufgrund der Laufzeiteigenschaften ($O(n)$) und einfachen Implementierbarkeit dieses Algorithmus (Jain et al. 1999, 278) wird er sehr häufig eingesetzt. Der Ablauf des klassischen K-Means Algorithmus (auch bekannt unter „Forgy's algorithm“[12]), ein iteratives Minimaldistanzverfahren, wird im Folgenden kurz skizziert. In der zugehörigen Abbildung 7.11 sind die Centroide durch ein + gekennzeichnet.

Algorithmus 3: Forgy's K-Means (nach Steinbach et al. 2000, 4)

Select K points as the initial centroids. [Abb. 7.11 a]
repeat
    Assign all points to the closest centroid. [Abb. 7.11 b]
    Recompute the centroid of each cluster .
until *the centroids don't change. [Abb. 7.11 c]*

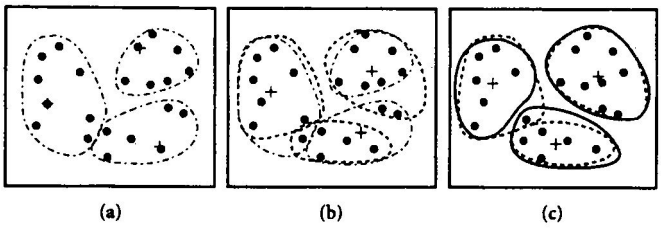


Abbildung 7.11: Schritte im Erstellen einer Cluster-Lösung beim K-Means Verfahren (Han und Kamber 2001, 350)

Bessere Lösungen werden erzielt, wenn die Centroide sofort neu berechnet werden, sobald ein Objekt seine Clusterzugehörigkeit ändert (continuous center adjustment, derart formuliert in der K-Means Fassung von MacQueen[13]). Gestützt auf die Ergebnisse von Larsen und Aone (1999) führten deshalb Steinbach et al. ihre Vergleichsuntersuchung mit dem modifizierten K-Means Algorithmus durch (Steinbach et al. 2000, 8), auf die in Kapitel 8.2 eingegangen wird.

Ein Nachteil des K-Means Verfahrens liegt in seiner Beeinflussbarkeit durch Ausreißer in den Ausgangsdaten: „Moreover, it is sensitive to noise and outlier data points since a small number of such data can substantially influence the mean value.“ (Han und Kamber 2001, 350) Der K-Medoid Algorithmus weist ein stabileres Verhalten auf.

[12] Forgy, E. (1965): Cluster analysis of multivariate data: Efficiency versus interpretability of classification. Biometrics, 21, 768-780.

[13] MacQueen, J. (1967): Some Methods for Classification and Analysis of Multivariate Observations. In: Lecam, L.M., Neyman, J. (eds.): Proc. 5th Berkely Symp. Math. Stat. Prob. 1965/66, Berkely 1967, 1 281–297

7.2.3 K-Medoid – eine auf Repräsentanten basierende Technik

Bei diesem Verfahren wird statt eines Centroids ein tatsächlich vorhandenes Datenobjekt, das am zentralsten im jeweiligen Cluster liegt (Medoid), als Cluster-Repräsentant bei der Berechnung des Gütekriteriums eingesetzt. Der Ablauf des K-Medoid Algorithmus lautet:

> „The basic strategy of k-medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the medoid) for each cluster. Each remaining object is clustered with the medoid to which it is the most similar. The strategy then iteratively replaces one of the medoids by one of the non-medoids as long as the quality of the resulting clustering is improved.“ (Han und Kamber 2001, 351)

Der K-Medoid Algorithmus ist gegenüber Ausreißern in den Ausgangsdaten unempfindlicher, als das K-Means Verfahren, ist aber aufwändiger zu berechnen. Bekannte algorithmische Umsetzungen des K-Medoid Verfahrens sind PAM (Partitioning around Medoids) und CLARA (Clustering LARge Applications) sowie CLARANS (Clustering Large Applications based upon RANdomized Search). Diese Algorithmen können größere Datenmengen (wie sie z.B. im Data Mining anfallen) effizienter bearbeiten (vgl. Han und Kamber 2001, 353 f.).

7.2.4 Bisecting K-Means

Dieser Algorithmus kann sowohl zur Generierung flacher, als auch hierarchischer Partitionierungen eingesetzt werden (im zweiten Fall geht er dann divisiv vor). Er weist ein lineares Laufzeitverhalten bezüglich der Anzahl der Eingabedaten auf und ist daher sehr effizient. Der Algorithmus arbeitet wie folgt (siehe Algorithmus 4):

Algorithmus 4: bisecting K-Means (Steinbach et al. 2000, 8)

```
repeat
    Pick a cluster to split.
    for ITER times do
        Find 2 sub-clusters using the basic K-Means algorithm (bisecting step)
    end
    take as result the split that produces the clustering with the highest overall
    similarity
until the desired number of clusters is reached.
```

Zum Ermitteln des Clusters, der als nächstes aufgespalten werden soll, kann man beispielsweise entweder generell den größten Cluster auswählen oder den Cluster, der durch Aufspaltung die Gütefunktion am positivsten beeinflusst.

7.3 Probabilistische Verfahren

Ein Beispiel für ein probabilistisches Fusionierungsverfahren (oder Optimierungsverfahren) sind Mischverteilungsverfahren. Ihnen liegt die Vorstellung zu Grunde, dass ein Objekt aus einer von mehreren Verteilungen stammt. In Abbildung 7.12 stellen die obere und mittlere Kurve zwei (Normal-)Verteilungen der Variablen von Cluster A (oberste Kurve) und Cluster B dar mit jeweils unterschiedlichen Mittelwerten (μ) und Standardabweichungen (σ), was an den unterschiedlichen Formen der Normalverteilungen zu erkennen ist. Diese beiden Normalverteilungen (= zu ermittelnde Cluster) sind jedoch nicht bekannt, stattdessen ist nur die Summe der Mischverteilungen bekannt (unterste Kurve), die durch die Ausgangsdaten gegeben ist. Die Parameter μ und σ werden bei den Mischverteilungsverfahren algorithmisch näherungsweise ermittelt: „Traditional approaches to this problem involve obtaining (iteratively) a maximum likelihood estimate of the parameter vectors of the component densities.“ (Jain et al. 1999, 280)

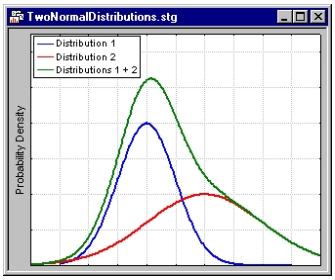


Abbildung 7.12: Beispiel für eine Mischverteilung (Quelle: `http://www.statsoft.com/textbook/graphics/Gclu1.gif`, Verifizierungsdatum: 10.10.2004, 22:55 Uhr MEZ)

Der *Expectation Maximization* (EM) Algorithmus stellt eine Implementierung dieser Grundidee dar: „In the EM framework, the parameters of the component densities are unknown, as are the mixing parameters, and these are estimated from the patterns.“ (Jain et al. 1999, 280) Die Parameter μ und σ der Cluster A und B werden beim EM-Algorithmus derart ermittelt, dass die Wahrscheinlichkeit für die bekannte Verteilung der Ausgangsdaten maximiert wird (vgl. Statsoft, o.J.). Für jedes Objekt wird die Zugehörigkeit zu einer Klasse auf Basis von (bedingten) Wahrscheinlichkeiten berechnet (expectation step). Dann werden die Parameter der Mischverteilung mittels der Zugehörigkeitswahrscheinlichkeiten der Objekte ermittelt (maximization step). Die letztendlich berechneten Zugehörigkeitswahrscheinlichkeiten geben die Klassenzugehörigkeit einer Instanz an. Für eine formale und detailliertere Beschreibung wird auf Witten und Frank (2000, 218 ff.) verwiesen.

Wie bei den partitionierenden Verfahren muss die Anzahl der zu bestimmenden Cluster vorgegeben werden. Außerdem wird von einer Unabhängigkeit der Attribute ausgegangen, die nicht zwangsläufig gegeben ist. Der probabilistische Ansatz

wird z.B. als reiner EM-Algorithmus im WEKA-Paket umgesetzt und in einer komplexeren Fassung im Programm Autoclass-C (siehe Kapitel A.4). Das letztgenannte Programm ermöglicht es sogar, unterschiedliche Verteilungsarten für die Attribute auszuwählen, da „[...] the normal distribution is usually a good choice for numeric attributes, [but] it is not suitable for attributes (such as weight) that have a predetermined minimum (zero, in the case of weight) but no upper bound, and in this case a 'log-normal' distribution is more appropriate." (Witten und Frank 2000, 224) Diese „log-normal" Verteilung ist z.B. für Termgewichte angemessen, bei denen das Nicht-Vorhandensein mit dem Wert 0 und das Vorhandensein mit einem nach oben offenen Wert bewertet wird.

7.4 Shared Nearest Neighbor Verfahren

Ausgehend von der Beobachtung beim Clustern von Dokumenten, dass bei hierarchischen Verfahren innerhalb eines Clusters mehrere Themengebiete vermischt sind und nicht in getrennten Klassen liegen, wurden Ertöz et al. (2003a) dazu angeregt, einen anderen Ansatz zu wählen, der ihren Angaben nach eine bessere Clusterqualität liefert[14]. Das häufig eingesetzte Cosinus-Maß zur Distanzberechnung erweist sich, so Ertöz et al., nicht immer als geeignet, was bei hierarchischen Clustering-Algorithmen eine schlechte Clusterqualität nach sich zieht:

> „For example, for the LA1 document set, a document's closest neighbor actually belongs to a different class 20% of the time. In such a scenario, hierarchical methods make many mistakes initially, and these mistakes can never be corrected, at least with standard hierarchical techniques." (Ertöz et al. 2003a, 88) (vgl. Kapitel 7.1.1)

Der Algorithmus von Ertöz et al. basiert auf einem „shared nearest neighbor clustering algorithm" (SNN), der ursprünglich von Jarvis und Patrick formuliert wurde[15]. Zur Grundidee dieser Art von Distanzberechnung vgl. Kapitel 6.6. Ertöz et al. beschreiben die Arbeitsweise ihres Ansatzes folgendermaßen:

> „The method [...] finds communities of documents, where a document in a community shares a certain fraction of its neigbors with at least some number of neighbors. While the probability of a document belonging to a class different from its nearest neighbor's class may be relatively high, this probability decreases as the two documents share more and more neighbors." (Ertöz et al. 2003a, 90)

[14] „Our goal was to find an algorithm that would more consistently produce clusters of documents with strong coherent themes [...]" (Ertöz et al. 2003a, 84)

[15] R.A. Jarvis and E.A Patrick (1973): Clustering Using a Similarity Measure Based on Shared Nearest Neighbors. IEEE Transactions on Computers, Vol. C-22, No. 11, November 1973

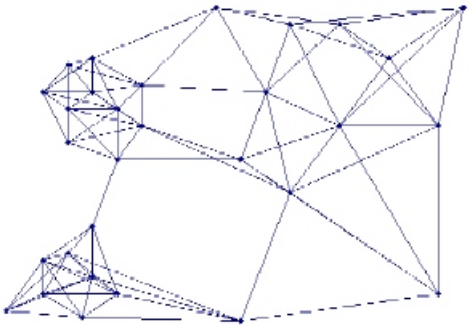


Abbildung 7.13: „nearest neighbor“-Graph (Ertöz et al. 2002, 7)

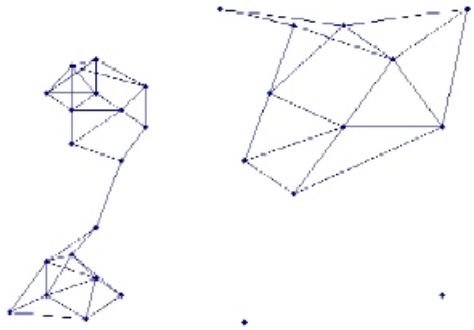


Abbildung 7.14: Ungewichteter „shared nearest neighbor“-Graph (Ertöz et al. 2002, 7)

Ausgehend von einer Ähnlichkeitsmatrix (z.B. berechnet unter Anwendung des Cosinus-Maßes), werden zunächst die *n* nächsten Nachbarn für jedes Dokument ermittelt. Im „nearest neighbor“-Graph sind die Dokumente *i* und *j* miteinander verbunden, wenn sie beide nächste Nachbarn zueinander sind (transitive Beziehung) (vgl. Abbildung 7.13). Im anschließend berechneten „shared nearest neighbor“-Graph besteht eine Verbindung zwischen *i* und *j*, wenn eine Kante im „nearest neighbor“-Graph die Dokumente *i* und *j* verbindet (vgl. Abbildung 7.14). Als Gewicht erhält diese Kante die Anzahl aller gemeinsamen Nachbarn von *i* und *j*. Ist das Gewicht einer Kante größer als ein zuvor festgelegter Schwellenwert, wird diese Verbindung als „strong link“ bezeichnet. Der weitere Ablauf ist in Algorithmus 5 beschrieben.

Algorithmus 5: SNN (nach Ertöz et al. 2003a, 89)

1.) Berechne für jedes Dokument i die „connectivity“ $conn[i]$ und die Anzahl der „strong links“

2.) Verwirf Dokument i falls gilt: $conn[i] <$ noise threshold (da es nur zu wenigen Nachbarn ähnlich ist). ODER: Falls $conn[i] >$ topic threshold, dann verwende Dokument i als Repräsentant für die Umgebung (da es zu vielen seiner Nachbarn Ähnlichkeit aufweist).

3.) Fusioniere alle Paare, die bei einem paarweisen Vergleich von Dokument (i, j) ein größeres Gewicht der verbindenden Kanten aufweisen, als durch den Wert merge threshold gefordert und wenn eines der Dokumente als Repräsentant dient.

4.) „Labeling step:“ Dokumente, die wegen des Wertes merge threshold nicht berücksichtigt wurden, werden einem Cluster zugeordnet. Dazu werden alle „shared nearest neighbor“ Listen aller Dokumente, die einen Cluster bilden, überprüft, ob bislang nicht zugeordnete Dokumente vorhanden sind und ob deren Gewicht der Verbindungskante größer als der zuvor definierte Schwellenwert von labeling threshold ist.

Die Clusteranzahl wird beeinflusst durch die Art der Ausgangsdaten und die Parameter wie z.B. die Größe der Shared-Nearest-Neighbor Liste; sie kann jedoch nicht exakt festgelegt werden. In den Experimenten in Kapitel 8.4 variiert die Anzahl der gefundenen Cluster in den verschiedenen Datensätzen – bei gleicher Parameter-Wahl – beträchtlich.

Das Laufzeitverhalten dieses Algorithmus wird als komplex ($O(n^2)$) beschrieben. Ein weiterer Nachteil besteht darin, dass nicht sämtliche Instanzen geclustert werden. Instanzen, die auch nicht mittels vordefinierter Schwellenwerte (z.B. labelling threshold) einem Cluster zugeordnet werden konnten, werden in einem großen „Rest-Cluster“ zusammengefasst.

7.5 Weitere Verfahren

Die im Folgenden vorgestellten Verfahren sollen einen Überblick über weitere Fusionierungsverfahren zur Clusterbildung liefern. Eine Anwendung dieser Verfahren bei den Experimenten in Kapitel 8 fand nicht statt.

7.5.1 Fuzzy-Clustering

Traditionelle Clustering-Algorithmen weisen Instanzen einem Cluster fest zu; die Cluster sind disjunkt (hard clustering). Beim Fuzzy-Clustering wird hingegen jeder Instanz mittels einer Membership-Funktion ein Zugehörigkeitswert zu jeder vorhandenen Klasse zugeordnet. Je höher der Wert, desto stärker ist die Zugehörigkeit zu einem Cluster. Der Hauptunterschied zu den in Kapitel 7.3 angeführten probabilistischen Verfahren liegt darin, dass keine den Daten zu Grunde liegende Mischverteilung angenommen wird, sondern die Zugehörigkeit zu einem Cluster mittels der Membership-Funktion berechnet wird.

Um das durch Fuzzy-Clustering gewonnene Ergebnis in ein hartes Clustering (wie bei den partitionierenden Verfahren) umzuwandeln, legt man fest, dass der Zugehörigkeitswert einen bestimmten Schwellenwert überschreiten muss. Bekanntester Fuzzy-Algorithmus ist der „fuzzy c-means“ (FCM), der im Vergleich zu K-Means nicht so stark die Tendenz aufweist, in einem lokalen Minimum stecken zu bleiben (vgl. Jain et al. 1999, 281).

7.5.2 Dichtebasierte Verfahren

Die Grundidee von dichtebasierten Clustering-Verfahren fasst Bergmann wie folgt zusammen (Bergmann 2004, 5):

> „Cluster sind mit Beispielen dicht besetzte Regionen im Datenraum, die von anderen Clustern durch Regionen geringer Dichte getrennt sind."

Dichtebasierte Verfahren können Clusterstrukturen ermitteln, die eine unregelmäßige Form aufweisen (vgl. Abbildung 7.15), die z.B. von K-Means Algorithmen nicht ermittelt werden können. Dieses Verhalten rührt daher, dass „[a] cluster, defined as a connected dense component, grows in any direction that density leads." (Han und Kamber 2001, 363) Diese Eigenschaft sei ein guter Schutz gegen Ausreißer, die z.B. den K-Means Algorithmus stark beeinflussen. Dichtebasierte Verfahren skalieren gut, d.h. die Speicherplatz- und Rechenzeitanforderungen wachsen nicht exponentiell mit der Zahl der Eingabedaten, wie z.B. bei den hierarchischen Verfahren. Jedoch wird den dichtebasierten Verfahren eine schwere Interpretierbarkeit der Ergebnisse zugeschrieben. Algorithmisch umgesetzt wird dieses Prinzip z.B. durch DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points To Identify the Clustering Structure) oder DENCLUE (DENsity-based CLUstering) (Berkhin 2002, 18 f.). Siehe hierzu auch Han und Kamber (2001, 363 ff.).

Abbildung 7.15: Unregelmäßig geformte Cluster können mit dichtebasierten Verfahren ermittelt werden (Berkhin 2002, 18)

7.5.3 Grid-basierte Verfahren

Bei den Grid-basierten Verfahren wird als Datenstruktur ein Gitter benutzt, das in eine endliche Anzahl von Zellen aufgeteilt wird. Diese Zellen kann man sich in mehrere Ebenen geschichtet vorstellen (vgl. Abbildung 7.16). Sie bilden eine hierarchische Struktur, da eine Zelle in einer folgenden Ebene in eine oder mehrere Zellen aufgeteilt wird. Diese Aufteilungsstruktur wird anschließend zum Bilden der Cluster benutzt. Hauptvorteil, so Han und Kamber, ist die hervorragende Laufzeiteigenschaft ($O(n)$) der Algorithmen, die von der Anzahl der Datenobjekte unabhängig ist und nur von der Anzahl der Zellen abhängt (vgl. Han und Kamber 2001, 370 ff.).

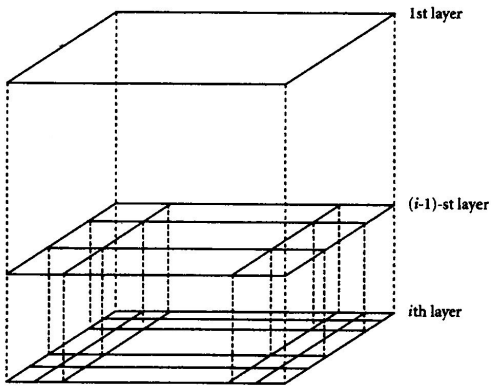


Abbildung 7.16: Beispiel für eine hierarchische Strukturierung bei Grid-basierten Fusionierungsverfahren (Han und Kamber 2001, 371)

Die positive Einschätzung der Eigenschaften von Grid-basierten Verfahren wird von Steinbach et al. nicht geteilt. Sie identifizieren folgende Problembereiche (vgl. Steinbach et al. 2000, 16):

- Die rechteckige Aufteilung kann die Clusterform nicht exakt nachbilden; eine Erhöhung der Zellenzahl zur besseren Approximation zieht eine schlechtere Performance nach sich.
- Bei hochdimensionalen Daten kann die Zahl der Zellen immens werden: „For example, even if each dimension is only split in two, there will still be 2^d cells. Given 30 dimensional data, a grid based clustering approach will use, at least conceptually, a minimum of a billion cells."
- Als Proximitätsmaße können ausschließlich die Minkowski-Metriken (L_1 und L_2) eingesetzt werden.

„Some typical examples of the grid-based approach include STING [STatistical INformation Grid], which explores statistical information stored in the grid cells; WaveCluster, which clusters objects using a wavelet transformation method; and CLIQUE [CLustering in QUEst], which represents a grid and density-based approach for clustering in high-dimensional data space." (Han und Kamber 2001, 370)

7.5.4 Inkrementelles Clustern

Beim inkrementellen Clustern (oder „conceptual clustering") werden nicht nur Cluster, sondern zusätzlich auch Beschreibungen ermittelt, die eine Klasse (oder ein Konzept) näher beschreiben. Die Qualität eines Clusters hängt somit nicht allein von den Ausgangsdaten, sondern auch von der Einfachheit und Abdeckungskraft der gefundenen Beschreibung ab. Die meisten inkrementellen Algorithmen verwenden zum Formulieren der Beschreibungen probabilistische Ansätze, wie der Algorithmus COBWEB, der in diesem Kapitel kurz skizzieren wird. (vgl. Han und Kamber 2001, 376)

COBWEB[16] erzeugt eine Hierarchie in Form eines Klassifikationsbaums (vgl. Abbildung 7.17). Die Knoten stehen dabei für ein Konzept, das durch die angegebenen Wahrscheinlichkeiten der Attributwerte beschrieben wird. Jeder Knoten mit seinen darunter liegenden Instanzen/Knoten stellt eine Partition dar.

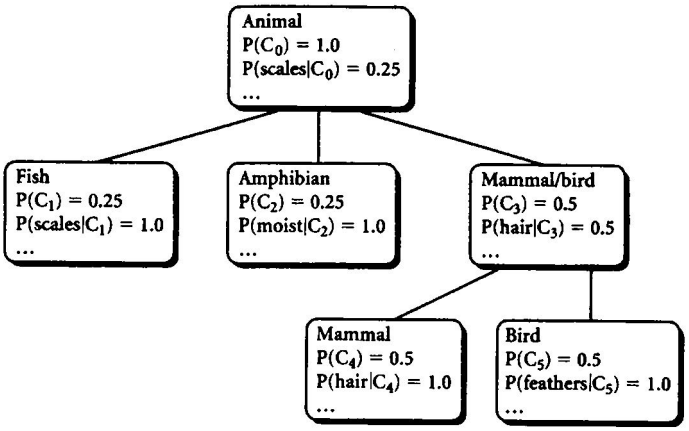


Abbildung 7.17: Klassifikationsbaum (Han und Kamber 2001, 377)

Um ein neues Objekt hinzuzufügen, wird der bestehende Klassifikationsbaum traversiert und dabei für jeden Knoten das Bewertungskriterium neu berechnet, um den geeigneten Platz (= Maximum des Bewertungskriteriums) zu ermitteln. Als Kriterium dient das Maß der *category utility* (siehe Han und Kamber 2001, 377 f.), das auf bedingten Wahrscheinlichkeiten basiert. Der Klassifikationsbaum ändert seine Gestalt mit jeder hinzugefügten Instanz: Bestehende Knoten erfahren eine Veränderung durch Aufspaltung oder Verschmelzen oder aber neue Konzepte werden in Form von weiteren Knoten ergänzt.

Die Vorteile von inkrementellen Verfahren liegen in ihrem nicht-iterativen Vorgehen und dem automatischen Ermitteln der optimalen Clusteranzahl. Han und Kamber beschreiben diese Verfahren als sehr aufwändig bezüglich Laufzeit und Speicherverbrauch:

> „Moreover, the probability distribution representation of clusters makes it quite expensive to update and store the clusters. This is especially so when the attributes have a large number of values since their time and space complexities depend not only on the number of attributes, but also on the number of values for each attribute.“ (Han und Kamber 2001, 379)

Als besonders nachteilig erweist sich außerdem die Abhängigkeit von der Reihenfolge der Eingabedaten: „An algorithm is order-independent if it generates the same partition for any order in which the data is presented.“ (Jain et al. 1999, 296) Dies

[16] vorgestellt in: Fisher, D. (1987): Knowledge aquisition via incremental conceptual clustering. Mach. Learn. 2, 139-172

trifft auf den COBWEB-Algorithmus und dessen Erweiterung CLASSIT (der für quantitative Daten metrischer Art geeignet ist) nicht zu. Außerdem wird bei der Berechnung der „category utility“ davon ausgegangen, dass die Attributwerte unabhängig voneinander sind, was beispielsweise bei Termen innerhalb eines Dokuments nicht zwangsläufig der Fall ist.

7.5.5 Künstliche Neuronale Netze

Die Künstlichen Neuronalen Netze (KNN) nehmen sich die Natur zum Vorbild: Wie im menschlichen Gehirn, in dem eine Vielzahl von Neuronen über Synapsen miteinander verbunden sind, sind hier ebenfalls Neuronen miteinander vernetzt. Sie kommunizieren mit anderen Neuronen durch Senden und Empfangen von Impulsen, die verschickt werden, wenn bestimmte Schwellenwerte für eine Aktivierungsfunktion überschritten werden. Eine mögliche Aktivierung geschieht durch Berücksichtigung aller Eingangsimpulse an einem Neuron sowie deren jeweilige Gewichtung. Unter den Gewichten versteht man die Parameter des Modells, die durch Lernen verändert werden und somit das Gesamtmodell beeinflussen (vgl. Mandl und Koelle 2001, 2 ff.).

KNN müssen den Spagat zwischen *Stabilität* einerseits und *Plastizität* andererseits schaffen. „The system is said to be stable if no pattern in the training data changes its category after a finite number of learning iterations.“ Werden aber neue Instanzen hinzugefügt, so soll sich ein KNN den Daten anpassen können, was mit Plastizität bezeichnet wird (vgl. Jain et al. 1999, 284). Stabilität ist für eine kontinuierliche, uniforme Clustereinteilung wünschenswert; die Plastizität soll zwecks Schaffung neuer Cluster nicht verloren gehen.

Self-organizing maps (SOM), entwickelt von Kohonen, können beispielsweise zum Clustern eingesetzt werden. Sie eignen sich außerdem dazu, hochdimensionale Daten in einem zwei oder dreidimensionalen Raum als „Karten“ zu visualisieren (siehe hierzu Abbildung 4.1 auf Seite 29) (vgl. Han und Kamber 2001, 381).

7.5.6 Evolutionäre Algorithmen

Bei den evolutionären Algorithmen werden natürliche Evolutionsprinzipien (Selektion, Rekombination und Mutation) nachempfunden, um mittels einer Population von Lösungsmöglichkeiten (d.h. eine Anzahl von gültigen Partitionen) auf die optimale Cluster-Lösung zu gelangen. Eine Fitness-Funktion beurteilt, ob eine Lösungsmöglichkeit (= ein Chromosom) für das Überleben in einer weiteren Generation ausgewählt werden kann. Hauptvertreter dieser Art von Algorithmen sind die Genetischen Algorithmen (GA), die hierbei am häufigsten zu Clustering-Zwecken eingesetzt wurden.

Ein Beispiel für eine Rekombinations-Operation ist die *Kreuzung*: Eine Kreuzung findet zwischen einem Paar von Chromosomen (den Eltern) statt und als Ergebnis erhält man ein neues Paar von Chromosomen (die Kinder) (Abbildung 7.18). Am Kreuzungspunkt (senkrechter Strich in der Abbildung) werden die Segmente der Eltern vertauscht. Bei der *Mutation* wird ein Chromosom an willkürlich gewählten Stellen verändert, so dass z.B. aus der Zeichenkette „11111110“ die „1**0**111110“ wird.

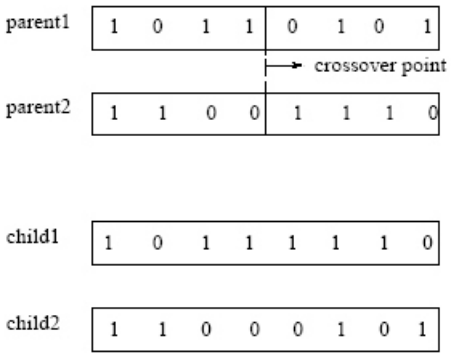


Abbildung 7.18: Kreuzung (Jain et al. 1999, 285)

Der Vorteil der Genetischen Algorithmen liegt in ihrer globalen Suche nach der optimalen Lösung. Die Kreuzungs- und Mutationsoperation können Lösungen erzeugen, die sich von den bisherig ermittelten völlig unterscheiden.

> „GAs represent points in the search space as binary strings, and rely on the crossover operation to explore the search space. Mutation is used in GAs for the sake of completeness, that is, to make sure that no part of the search space is left unexplored.“ (Jain et al. 1999, 285 f.)

Andere Fusionierungsalgorithmen versuchen ebenfalls eine extensive Suche, bleiben jedoch in einem lokalen Minimum stecken. Bei den GA sind die Sprünge im Lösungsraum viel größer und überraschender, was sich positiv auf die Ermittlung eines globalen Optimums auswirkt.

Das Hauptproblem der GA liegt in der Fülle der möglichen Chromosomen, die alle von der Fitness-Funktion bewertet werden müssen. Bei einer Variante der GA, dem „edge-based crossover“, beläuft sich die Laufzeit auf $O(K^6 + N)$ für N Objekte und K Cluster. Einen Ausweg stellen so genannte hybride Verfahren dar, bei denen GA zum Finden einer geeigneten Startpartition eingesetzt werden und die eigentliche Partitionierung von einem effizienten K-Means Algorithmus ermittelt wird. Dieses Verfahren liefert bessere Ergebnisse als GA alleine (vgl. Jain et al. 1999, 287).

7.6 Zusammenfassung

Es gibt zahlreiche Fusionierungsverfahren, um Cluster zu erzeugen. Sie weisen unterschiedliche Eigenschaften auf, die in den vorangegangenen Abschnitten beschrieben wurden. Eine Auswahl des Fusionierungsalgorithmus kann nur angesichts eines konkreten Anwendungskontexts getroffen werden, da es einen allgemein gültigen, optimalen Algorithmus nicht gibt. Sollen z.B. Dokumente geclustert werden, spielen ganz andere Faktoren bei der Auswahl eines Algorithmus eine Rolle als z.B. beim Clustern von Gensequenzen. Für den Bereich „Clustern von Dokumenten“ gibt es experimentelle Untersuchungen, die für diesen Anwendungsbereich verschiedene Fusionierungsverfahren einander gegenüberstellen, um dadurch ein gut geeignetes Verfahren für diesen Zweck zu identifizieren (siehe Kapitel 8.2).

8 Clustering-Experimente mit Patentdaten

Im Rahmen dieser Arbeit werden Clustering-Experimente durchgeführt, die von Nutzern bewertet werden, um für den hier vorliegenden Anwendungsbereich der Patentrecherche und -information möglichst geeignete Verfahren zur Cluster-Bildung zu ermitteln. Dafür werden zunächst Annahmen formuliert, die im weiteren Verlauf der Arbeit experimentell bestätigt oder widerlegt werden:

Annahme 1: *Das Entfernen von Patentfamilien-Doppeln in den Ausgangsdaten führt zu einer besseren Clusterqualität.*[1]

Annahme 2: *Ein Verfahren zur Erzeugung von Clustering-Lösungen sticht mit qualitativ hochwertigen Lösungen deutlich hervor.*

Annahme 3: *Die Gruppierung von Patentdokumenten mittels der IPC-Klassen ist per se ideal.*

Für die Versuche wird ausschließlich frei verfügbare Software eingesetzt. Es handelt sich um das Programm CLUTO, den SNN-Algorithmus, den EM-Algorithmus (implementiert im WEKA-Paket) und um das Programm Autoclass-C. Die Herkunfts- und Lizenzinformationen, sowie eine Kurzbeschreibung der Programmfähigkeiten und der Formate der Eingabedaten der einzelnen Software-Programme befinden sich im Anhang (Anhang A).

8.1 Datengrundlage

In den folgenden Kapiteln werden die Schritte zur Auswahl und Aufbereitung der Patentdaten beschrieben, um mit den zuvor genannten Software-Tools Clustering-Lösungen zu erzeugen.

8.1.1 Vorgehen zur Aufbereitung der Daten aus der Patentdatenbank PATDPA

Die Patentdokumente, die auf eine Anfrage an die Datenbank PATDPA über den STN-Host als Ergebnis zurückgeliefert werden, werden im „brief“-Format ausgegeben und

[1] Diese Annahme wurde durch Beobachtungen während der Datenaufbereitung (vgl. Kapitel 8.1.1) motiviert.

in einer Textdatei gespeichert. Diese „Rohdaten“ müssen entsprechend den Anforderungen der zur Durchführung des Clusterings eingesetzten Software (CLUTO, WEKA, SNN und Autoclass-C) in ein spezielles Eingabeformat konvertiert werden. Dies erledigt die im Zuge der Magisterarbeit erstellte JAVA-Klasse `PatentPreprocessing.java`.

Die Inhalte der Datenbankfelder `TI` (title), `AB` (abstract) und `MCLM` (Mainclaim) eines Patentdokuments werden als Ausgangsdaten verwendet. Mit Hilfe der hierfür weiterentwickelten JAVA-Klasse `PatentParser.java`[2] werden diese Felder aus dem Datenbank-Ausgabeformat extrahiert. Zu Beginn des TI-Feldes ist in Klammern die Art des Dokuments notiert, z.B. „(B1) Hauptanspruch einer EP-Patentschrift“ (weitere Kürzel siehe Thomä und Tribiahn 2002, 55) und im laufenden Text des Mainclaims wird mit Zahlen in Klammern auf Teile einer Zeichnung referenziert. Diese geklammerten Ausdrücke wurden mittels Regulärer Ausdrücke entfernt, da sie ansonsten als Terme zur Beschreibung eines Dokuments herangezogen wurden und dadurch eventuell einen ungünstigen Einfluss auf die Berechnung der Cluster ausgeübt hätten.[3]

Anschließend wurden Stoppwörter entfernt. Die Stoppwortliste[4] wurde um Terme ergänzt, die auf Grund eigener Beobachtungen sehr häufig in den Patentdokumenten vorkamen[5]. Terme, die aus weniger als drei Zeichen bestehen, werden nicht zur Beschreibung eines Dokuments herangezogen.

Anschließend wurden die Ausgangsdaten mittels Stemming auf ihre Grundform reduziert, wozu der Snowball-Stemmer[6] eingesetzt wurde, der ein regelbasiertes Verfahren zum Abtrennen der Suffixe verwendet. Alle Patentdokumente mit einer Gesamtzahl an Termen ≥ 5 (nach Stoppwort-Entfernung und Stemming) wurden weiterverarbeitet. Es erfolgte eine Termgewichtung nach dem Okapi BM25 Schema mit den Parametern $k_1 = 1.5$ und $b = 0.8$. Die Parameterwahl lehnt sich an die von Kamps et al. (2004, 3) durchgeführten Versuchen mit Web-Dokumenten im Rahmen von TREC 2003 (Web-Track) an. Alle Terme wurden ohne Berücksichtigung des Standortes in-

[2] Die Original-Klasse entstand als Teilprojekte der studentischen Gruppe (1a) anlässlich des Projekt-Seminars „Semantic Web und Ontologien“ (Wintersemester 2003/2004, Universität Hildesheim) unter der Leitung von Diplom-Informationswissenschaftler Robert Strötgen, Dipl.-Inform. Ralph Koelle und Dr. René Schneider.

[3] Der Reguläre Ausdruck zur Entfernung der Zeichnungs-Referenznummern weist in der gegenwärtigen Fassung den Nachteil auf, im Text vorhandene chemische Formeln zu verändern, bspw. wird aus Mg(NO3)2 x 6H2O+LiNO3 nach Anwendung des Regulären Ausdrucks MgNO x HOLiNO. Zudem bleiben bei Verwendung von alphanumerischen Referenzen einzelnen Buchstaben nach der Bereinigung übrig, z.B. wird aus „Verstaerker (v2) fuer das Signal“ im Ergebnis „Verstaerker v fuer das Signal“. Dies wird in den weiteren Verarbeitungsschritten berücksichtigt.

[4] Liste von 603 Stoppwort-Tokens im Deutschen, erstellt von der Universität Neuchâtel (CH) im Rahmen der Teilnahme an CLEF. Stoppwortliste heruntergeladen von http:\\www.unine.ch/info/clef/germanST.txt (Verifizierungsdatum 05.07.2004, 22:09 Uhr MEZ).

[5] Stoppwortliste ergänzt um: derzeit, enthalten, enthaelt, enthält, fuer, gemaess, Geraet, Gerät, Geraete, Geräte, hinsichtlich, jeweil, jeweilige (-m, -n, -s), jeweils, Methode, Methoden, verfahren, vorricht, Vorrichtung, Vorrichtungen, waehrend, waere, waeren, wobei, wodurch, wofuer, wofür, wovon

[6] `http://snowball.tartarus.org`, Verifizierungsdatum: 05.10.2004, 10:14 Uhr MEZ

nerhalb eines Patentdokuments gleichermaßen gewichtet, d.h. Terme im Titel eines Dokuments erhielten kein besonderes Gewicht. Da die Gewichtung nach dem Okapi BM25 Schema eine Normalisierung bezüglich der Dokumentenlänge beinhaltet (siehe Kapitel 5.3.2), wurde auf eine weitergehende Normalisierung oder Standardisierung (vgl. Kapitel 5.4) verzichtet.

Bei den Vorab-Tests ließ sich beobachten, dass ein Patentdokument mit identischem Titel zwei Mal im gleichen Cluster erschien. Bei genauerer Untersuchung der Patentdokumente stellt sich heraus, dass z.B. das Patent mit dem Titel „Verfahren und Gerät zum Übersetzen von einer Sprache in eine andere“ unter den (eindeutigen, als ID fungierenden) Systemnumbers DE69712216.6 und EP97910114.4 in der Datenbank existiert, sich jedoch in den für die Experimente einbezogenen Datenfeldern nicht unterschieden (hier: gleicher Inhalt des Abstracts). Die Ursache für diese Beobachtung liegt in der dynamischen Fortschreibung der Datenbank PATDPA und der unterschiedlichen Herkunft der Patentanmeldungen (siehe Kapitel 3.2.4): Diese „doppelt“ aufgeführten Patente sind Mitglieder derselben Patentfamilie, fanden jedoch durch Anmeldung bei verschiedenen Organisationen (EP, WIPO) Eingang in die Datenbank PATDPA (zu erkennen im Länder-Kürzel „DE“ oder „EP“ der Systemnumber).

In der Annahme, dass diese „Patentfamilien-Doppel“ (PF-Doppel) die Cluster-Lösung und die Bewertung einer Lösung verzerren, werden die Experimente mit und ohne PF-Doppel durchgeführt (siehe Annahme 1, Kapitel 8). Dabei werden die PF-Doppel anhand ihres Titels identifiziert und bei völliger Übereinstimmung des Abstracts und/oder Mainclaims (ermittelt durch String-Vergleich) nur eines dieser Dokumente in die Ausgangsdaten für die Clustering-Experimente mit einbezogen. Geringe Unterschiede (z.B. Trennungsstriche innerhalb eines Wortes oder ein anderes Nummerierungs-Schema zum Referenzieren von textueller Beschreibung und Zeichnung) führten zur Aufnahme beider Dokumente in die Ausgangsdaten.

8.1.2 Datengrundlage für die Experimente

In diesem Kapitel werden die Auswahlkriterien für die Anfragen beschrieben, die im Zuge der Experimente von den Clustering-Verfahren verarbeitet werden sollen.

8.1.2.1 Auswahl der Anfragen

Grundlagen für eine Antwortmenge von Patentdaten stellt eine Anfrage an die Datenbank PATDPA dar. Diese Anfragen sollen das vage formulierte Informationsbedürfnis eines fiktiven Informationssuchenden widerspiegeln, der sich zu bestimmten Themenbereichen und den dort vorhandenen Patenten einen groben Überblick verschaffen will. Da auf Grund datenschutzrechtlicher und praktischer Gründe nicht

auf Original-Anfragen von Nutzern der Datenbank PATDPA des STN-Hosts zurückgegriffen werden konnte, mussten Anfragen selbst zusammengestellt werden.

Zunächst bezogen sich die Anfragen auf die gesamte Datenbank PATDPA, über alle IPC-Klassen hinweg, was sich jedoch beim Betrachten der Clustering-Ergebnisse als nicht sinnvoll erwies. Auf Grund der Breite des Themenspektrums, über das die Patentdokumente verstreut waren, muteten die erzeugten Cluster zusammenhanglos und wirr an[7].

Um thematisch koheräntere Cluster zu erzeugen, wurden die Suchanfragen auf eine Hauptklasse der IPC eingeschränkt. D.h., sämtliche Patentdokumente wurden von einem Menschen der hier verwendeten IPC Hauptgruppe G06F017 zugeordnet (Physik → Datenverarbeitung; Rechnen; Zählen → Elektrische digitale Datenverarbeitung → Digitale Rechen- oder Datenverarbeitungsanlagen oder -verfahren).

8.1.2.2 Auswahl der Datensätze für die Experimente

Um die Datengrundlage für die Clustering-Versuche festzulegen, wurde eine Statistik über die Anfragen des fiktiven Informationssuchenden erstellt (siehe Tabelle 8.1). Der Umfang der Antwortmenge auf eine Anfrage spiegelt den Stand der Datenbank PATDPA vom 22.08.2004 wider.

In Tabelle 8.1 ist aufgeführt, wie viele Dokumente als Treffer auf die Anfrage geliefert wurden. Das Messenger-System erlaubt eine klassische Boolesche Suche, die durch Proximitätsoperatoren wie dem hier angewandten „(S)“-Operator (Worte müssen im gleichen Satz auftreten) erweitert werden können. Neben der Größe der Treffermenge wird die Anzahl der Dokumente angegeben, bei denen nur folgende Felder mit Inhalt gefüllt sind: TI; TI und AB; TI, AB und MCLM; TI und MCLM. Die Mehrzahl der nachgewiesenen Patentdokumente enthält hauptsächlich Informationen aus den Feldern TI und AB (377/412 ohne PF-Doppel). Hebt man die in Kapitel 8.1.2.1 erwähnte Beschränkung auf, dass mindestens fünf Terme (nach Stoppwort-Elimination und Stemming) innerhalb eines Patentdokuments vorkommen müssen, so verschiebt sich die Anzahl. In diesem Falle weist die Mehrheit der Patentdokumente nur einen Titel auf (564 vs. 506 ohne PF-Doppel). Aufgrund dieser Feststellung werden letztlich nur Patentdokumente als Eingabedaten für die Clustering-Verfahren zugelassen, die eine Mindestlänge von fünf Termen aufweisen. Ohne diese Beschränkung wären zahlreiche Dokumente mit einbezogen worden, die nur aus wenigen Termen (bisweilen gar aus einem einzigen Term) bestehen und das Clustering-Ergebnis dadurch womöglich verzerrt hätten. Für die Experimente wurden letztendlich nur Anfragen ausgewählt, die aus mehr als achtzig Dokumenten bestehen (in Tabelle 8.1 durch # gekennzeichnet).

[7] Ermittelt durch eigene Testläufe und Betrachten der Lösungen in den Vorab-Tests.

Anfrage + G06F017/ICM	Gesamtzahl Dokumente (o. PF-D)	TI (m. PF-D)	TI/AB (m. PF-D)	TI/AB/MCLM (m. PF-D)	TI/MCLM (m. PF-D)	Gesamtzahl Terme (m. PF-D)	max. Anzahl Terme/Dok.	Durchschnittl. Anz. Terme/Dok. (m. PF-D)
# bild? (S) verarbeit? +	100/126	5/5	23/25	10/10	62/86	2554/2577	152	53/51
# brows? +	116/140	52/55	42/46	2/2	20/37	1381/1415	69	25/27
# datenuebertragung? +	102/124	10/13	46/54	8/8	38/49	1893/1927	82	39/39
daten? (S) komprimier? +	25/35	4/4	3/3	2/2	16/26	726/752	92	43/45
# digital? AND bild? +	81/96	10/10	26/26	8/8	37/52	2038/2047	152	47/49
index? (S) such? +	32/42	7/8	6/6	1/1	18/27	904/904	88	42/43
internet? AND such? +	47/50	9/9	32/34	2/2	4/5	1161/1162	108	42/42
# medizin? +	81/95	32/34	31/33	3/3	15/25	1308/1312	81	29/30
# multimedia? +	124/152	83/92	22/26	1/1	18/33	1325/1372	135	19/21
muster? (S) erkenn? +	10/14	0/1	5/8	0/0	5/5	451/452	102	57/46
# navig? +	94/108	43/46	38/39	1/1	12/22	1326/1331	81	25/27
objektorient? +	48/62	15/16	14/16	3/3	16/24	963/970	87	33/33
# server? AND client?	121/152	13/17	67/74	5/5	36/57	2123/2140	101	40/40
transfer? +	54/57	19/19	22/22	3/3	10/13	1211/1211	113	35/35
Summen	**1035/1253**	**302/331**	**377/412**	**49/49**	**307/461**			

Tabelle 8.1: Statistische Werte über Anfragen an die Datenbank PATDPA zur Ermittlung der Datengrundlage für Clustering-Versuche (o. PF-D = ohne Patentfamilien-Doppel, * = für die Experimente eingesetzte Anfrage)

8.2 Auswahl der Clustering-Verfahren

Die Datengrundlage für die Experimente in dieser Arbeit stellen Patentdokumente, d.h. Text-Dokumente dar. Da bislang keine experimentellen Untersuchungen speziell mit Patentdokumente als Ausgangsdaten verfügbar sind, muss für die Auswahl geeigneter Clustering-Verfahren auf Untersuchungen zum Thema „Clustern von Text-Dokumenten“ zurückgegriffen werden. In diesem Kapitel werden verschiedene Analysen und deren Ergebnisse in Kurzform vorgestellt, um darauf basierend eine Auswahl der Clustering-Verfahren für die Experimente im Zuge dieser Arbeit zu treffen.

Analysen zum Clustern von Text-Dokumenten und deren Ergebnisse

Hierarchische Verfahren galten lange Zeit in der Literatur zu Clustering-Verfahren den partitionierenden Verfahren als überlegen: "Nevertheless, there is the common belief that [...] partitional algorithms are actually inferior and less effective than their agglomerative counterparts.“ (Zhao und Karypis 2003, 2) Daher wurden bei durchgeführten Vergleichsanalysen häufig nur hierarchische Verfahren berücksichtigt, so z.B. in der Analyse von El-Hamdouchi und Willet (1989, 226), die darin die Verfahren „group average“ und „complete Linkage“ als am geeignetsten zur Bestimmung der inter-Cluster Proximität identifizierten. Erst in den vergangenen Jahren wurden partitionierende Verfahren und ihre Eignung zum Clustern von Dokumentenmengen „wiederentdeckt“ (vgl. hierzu Zhao und Karypis 2003), wozu die nachfolgend kurz vorgestellten Analysen beigetragen haben.

Steinbach et al. (2000) verglichen unter Verwendung von acht Datensätzen mehrere Clustering-Verfahren miteinander. Es handelt sich um die partitionierenden Verfahren K-Means und „bisecting K-Means“ (jeweils mit Refinement-Phase, vgl. Kapitel 7.2.1) sowie die hierarchischen Verfahren mit den intra-Cluster Proximitätsmaßen UPGMA (average Linkage), dem Centroid-Verfahren und einem dritten Verfahren. Die erzeugten Lösungen werden hinsichtlich ihrer Qualität mit einer existierenden Lösung (siehe hierzu Kapitel 9.1) verglichen. Von den hierarchischen Verfahren erweist sich das UPGMA-Verfahren als das Beste, so dass nur dieses mit den partitionierenden Verfahren K-Means und „bisecting K-Means“ verglichen wurde. Als Ergebnis ermitteln Steinbach et al. (2000, 14):

- „Bisecting K-Means“ ist besser als K-Means und das UPGMA-Verfahren. Liefern andere Verfahren bessere Ergebnisse, dann ist „bisecting K-Means“ nur geringfügig schlechter.
- Das K-Means Verfahren ist generell besser als UPGMA, obwohl es schlechter als das „bisecting K-Means“ Verfahren ist.

Einen umfassenden Vergleich von hierarchischen und partitionierenden Verfahren zum Clustern von Dokumenten führten Zhao und Karypis durch. In zwei Artikeln,

Zhao und Karypis (2002) und Zhao und Karypis (2003), veröffentlichten sie die Ergebnisse ihrer Analyse, bei der sie als partitionierendes Verfahren den „bisecting K-Means“ Algorithmus mit sechs unterschiedlichen Gütefunktionen (jeweils mit Refinement-Phase) und ein hierarchisch-agglomeratives Verfahren mit neun unterschiedlichen Maßen zur Bestimmung der inter-Cluster Proximität anhand von jeweils zwölf Datensätzen miteinander verglichen. Die Qualität der ermittelten Lösungen wurde durch Vergleich mit einer existierenden Lösung bestimmt. Außerdem wurde ein drittes Verfahren, das *constrained agglomerative Clustering Verfahren* in die Vergleichsuntersuchung mit einbezogen.

Die Grundidee eines „beschränkt“ (engl. = constrained) arbeitenden Verfahrens liegt darin, dass mittels eines partitionierenden Algorithmus Initial-Cluster gefunden werden, auf die anschließend jeweils ein hierarchischer Algorithmus angewandt wird. Im letzten Schritt werden die Teile in eine hierarchische Lösung überführt. Durch das Kombinieren erhofft man sich eine bessere Gesamtlösung, da die globale Sichtweise auf die Dokumentenkollektion mittels der Gütefunktion der partitionierenden Verfahren und die lokale Sichtweise über die (Un-)Ähnlichkeit mittels der hierarchischen Verfahren hierbei zusammengeführt werden (vgl. Zhao und Karypis 2003, 7).

Folgende Ergebnisse lassen sich aus den Versuchen von Zhao und Karypis (2003, 10 f.) ableiten:

- Partitionierende Verfahren sind sämtlichen hierarchisch-agglomerativen Verfahren überlegen.
- Bei den hierarchischen Verfahren liefert das UPGMA-Verfahren (Average Linkage) zur Bestimmung der inter-Cluster Proximität die besten Ergebnisse.
- Bei den partitionierenden Verfahren wird die überwiegende Mehrzahl der besten Ergebnisse unter Anwendung der Gütefunktion $\mathcal{I}_2$ erzeugt. Auf Platz zwei liegt die Gütefunktion $\mathcal{H}_2$ (vgl. Kapitel 7.2.1).
- Das „constrained agglomerative Clustering“ erzeugt eindeutig bessere Ergebnisse im Vergleich zu den vom hierarchisch-agglomerativen Verfahren erzeugten Lösungen. Vergleicht man die Ergebnisse dieses Verfahrens mit denen des partitionierenden Verfahrens, so ergibt sich kein eindeutiges Bild der Überlegenheit.

Zhao und Karypis untersuchten in ihrem jüngst veröffentlichten Forschungsbericht, ob zum Erstellen von disjunkten Clustern die Anwendung von Fuzzy-Clustering Verfahren Vorteile bringt. Als Ergebnis formulieren sie: „Our experimental results and analysis show that the soft criterion functions tend to consistently improve the separation between the clusters, and lead to better clustering results for most datasets.“ (Zhao und Karypis 2004, 9) Da aber keine frei verfügbare Implementierung dieses Verfahrens vorhanden ist, konnten im Zuge dieser Arbeit keine Experimente durchgeführt werden, die die Prinzipien von Fuzzy-Clustering verwenden.

Implikationen für die Auswahl von Verfahren im Rahmen dieser Arbeit

Auf Basis dieser Ergebnisse wurden für die Experimente im Rahmen der Magisterarbeit das **„bisecting K-Means-Verfahren“** (mit Refinement-Phase und der Gütefunktion $\mathcal{I}_2$, implementiert in der Software CLUTO) **als Vertreter eines partitionierenden Verfahrens** ausgewählt. Hierarchische Verfahren werden für die Experimente nicht berücksichtigt, da sie sich in den angeführten Analysen als unterlegen erwiesen haben. Das von Zhao und Karypis (2002) vorgeschlagene „constrained agglomerative Clustering“ Verfahren wird auf Grund der uneinheitlichen Ergebnisse in den Analysen und mangels einer verfügbaren Implementation ebenfalls nicht in die Experimente mit einbezogen. Außerdem wurde das **SNN-Verfahren** (vgl. Kapitel 7.4) ausgewählt, da es im Artikel von Ertöz et al. (2003a) und den dort durchgeführten Versuchen mit vielversprechenden Eigenschaften beschrieben wurde („Our research indicates that clustering based on shared nearest neighbors is a better approach than K-means clustering for finding groups of documents with a strong, coherent topic or theme.“ (Ertöz et al. 2003a, 100)). Zuletzt soll noch ein **probabilistisches Verfahren** zur Generierung von Cluster-Lösungen eingesetzt werden (realisiert in der Autoclass-C Software und dem WEKA-Paket), um Aussagen über die Eignung dieser Verfahrensgruppe zu treffen. Insgesamt werden drei unterschiedliche Fusionierungsverfahren untersucht und ihre erzeugten Lösungen beim Clustern von Patentdokumenten durch menschliche Juroren miteinander verglichen.

8.3 Beobachtungen in den Vorab-Versuchen

In Vorab-Versuchen zu den Experimenten wurde das Verhalten der verschiedenen Verfahren für Daten des Anwendungsbereichs „Patente“ ermittelt. Hierbei wurde das Ziel verfolgt, mögliche Parameter zur Feineinstellung der Verfahren zu ermitteln. Die Vorab-Versuche und die Experimente wurden mittels des im Zuge dieser Magisterarbeit entwickelten Programms `ExperimenterGUI` durchgeführt, das im Anhang in Kapitel B.2 näher beschrieben wird. Dieses Programm reicht die visuell am Bildschirm gewählten Parameter an die Programme zur Cluster-Erzeugung weiter und ermöglicht ein sofortiges Betrachten der ermittelten Clustering-Lösungen, was das experimentelle Vorgehen zur Ermittlung der Parameter stark vereinfacht.

Der im **WEKA-Paket** (siehe Anhang A.2) **implementiere EM-Algorithmus** erwies sich in diesen Vorab-Tests als gänzlich ungeeignet für die Aufgabenstellung. Ausschlaggebend dafür war die Verteilung der Instanzen über die Cluster: Wählte man die automatische Ermittlung der Clusteranzahl, so fand der WEKA EM-Algorithmus maximal 1–4 Cluster, von denen die Mehrzahl der Instanzen in einem einzigen, sehr großen Cluster lag. Beispielsweise wurden die 102 Patentdokumente auf die Anfrage „bild? (S) verarbeit?“[8] in zwei Cluster der Größe 100 und 2 aufgeteilt. Wurde

[8] einschl. PF-Doppel, Gewichtung nach TF/IDF, Mindestlänge eines Dok. = 1 Term

die Clusterzahl im Vorfeld entsprechend der von Autoclass-C automatisch ermittelten Anzahl festgelegt, so bildete der WEKA EM-Algorithmus einen großen Cluster und wenige Cluster mit je nur einem Element. Autoclass-C erzeugte dagegen eine ausgewogenere Verteilung. Im zuvor genannten Beispiel ermittelte (für die fixe Clusteranzahl 10) der WEKA EM-Algorithmus als Ergebnis Cluster der Größen 1, 1, 1, 1, 92, 1, 1, 2, 1, 1, während Autoclass-C Cluster der Größe 17, 15, 12, 10, 10, 10, 8, 8, 8, 4 berechnete. Hierbei ist besonders zu bemerken, dass der in WEKA realisierte EM-Algorithmus nicht einmal die vorhandenen Patentfamilien-Doppel in der erzeugten Lösung identifizierte und zusammenfasste. Dasselbe Verhalten zeigte sich bei sämtlichen anderen Anfragen, weshalb der EM-Algorithmus von WEKA keine Verwendung bei den Experimenten fand.

Beim **SNN-Algorithmus** können sechs verschiedene Parameter gleichzeitig variiert werden. Je nach Parameterwahl erhält man entweder viele Cluster, die nur aus einem Dokument bestehen oder nur ganz wenige, sehr große Cluster. Der Algorithmus ist insgesamt schwer zu parmeterisieren: Leichte Veränderungen an den Parametern (z.B. ± 0.05) erzeugen ein vollkommen anderes Ergebnis, wobei die Clusteranzahl und die Anzahl der Instanzen pro Cluster stark variiert. Auch bei Anwendung derselben Parameter auf verschiedene Datensätze ist dieses Verhalten zu beobachten. Da in dieser Arbeit die Eignung des SNN-Verfahrens für Patent-Dokumente tendenziell eingeschätzt werden soll, wird auf umfangreiche Untersuchungen zur optimalen Parameterkonstellation verzichtet (was z.B. ein Bewerten der Ergebnisse durch Juroren bedeutet hätte).

8.4 Durchführung der Experimente

Das Programm **CLUTO** wurde zur Ermittlung von Clustering-Lösungen mittels des **„bisecting K-Means“**-Verfahrens eingesetzt. Als Parameter wurden gewählt: `-ntrials = 300` (Anzahl der Lösungsalternativen, die berechnet werden), `-niter = 15` (Anzahl der „Refinement“-Iterationen, vgl. Kapitel 7.2.1), `-seed = 75` (Initialwert des Zufallsgenerators) und `-cstype = best` (Auswahl, welcher Cluster aufgesplittet werden soll, vgl. Kapitel 7.2.4). Eine Erhöhung des Parameters `-ntrials` geht mit einem größeren Zeitaufwand zur Bestimmung der Gesamtlösung einher, wobei durch stichprobenartigen Vergleich kein Unterschied in den erstellten Clustern festzustellen war (bis auf vereinzelt anders zugeordnete Instanzen; die Grobeinteilung blieb bestehen). Die CLUTO-interne Möglichkeit zur Termgewichtung nach TF/IDF wurde nicht genutzt (`-colmodel= none`), da für alle Verfahren einheitlich das Okapi-Gewichtungsschema verwendet werden soll. Als Ähnlichkeitsmaß wurde das Cosinus-Maß (`-sim = cos`) eingesetzt.

Da bei partitionierenden Verfahren die Zahl der zu ermittelnden Cluster im Voraus angegeben werden muss, wird zur Festlegung der Clusteranzahl wie folgt verfahren:

Anfrage	Anz. Dok. / (mit PF-D)	probabil. Verf. / (mit PF-D)	SNN / (mit PF-D)	SNN einelementig / (mit PF-D)	bisecting K-Means / (mit PF-D)
bild? (S) verarbeit?	100/126	10/12	22/42	11/24	10/13
brows?	116/140	12/13	49/54	37/36	12/14
datenuebertragung?	102/124	10/11	24/36	16/18	10/12
digital? AND bild?	81/96	10/11	21/30	13/17	8/10
medizin?	81/95	9/9	24/28	13/13	8/10
multimedia?	124/152	12/14	40/59	25/36	12/15
navig?	94/108	11/9	40/43	28/31	9/11
server? AND client?	121/152	12/14	47/58	37/37	12/15

Tabelle 8.2: Anzahl der erzeugten Cluster

Die kaufmännisch gerundete Zahl von 10% der Gesamtanzahl der Dokumente wird als Clusteranzahl verwendet. Die im Zuge dieser Arbeit formulierte Faustregel deckt sich größtenteils mit der von Autoclass-C ermittelten „optimalen“ Clusteranzahl (siehe Tabelle 8.2).

Autoclass-C führte bei den vorliegenden Daten (Term nicht vorhanden = 0, nach oben offener Wert bei Vorhandensein je nach Gewichtung) zwangsweise eine Normalisierung durch. Dazu wird für jedes Attribut *log(Attributwert - Nullpunkt des Attributs)* berechnet und auf die Gauss'sche Normalverteilung abgebildet (vgl. hierzu Dokumentation zum Autoclass-C Paket, Datei: preparations-c.txt). Außerdem wird angenommen, dass die Attribute unabhängig voneinander sind, was bei Termen aus Texten bedingt zutrifft. Die Anzahl der maximalen Schritte zum Ermitteln der (näherungsweise) optimalen Parameter der Mischverteilung wird auf 300 Iterationen (`max_n_tries = 300`) festgelegt; beim Überschreiten dieser Anzahl wird mit der Suche abgebrochen.

In den Experimenten wurde der **SNN-Algorithmus** mit folgenden Parametern gestartet: Größe der Nearest-Neighbor Liste = 24 (`NN = 24`), Anzahl der Strong-Links = 30% der Eingabedaten (`strong = 0.3`), Anzahl der Representative Points = 70% der Links innerhalb des Nearest-Neighbor Graphen (`topic = 0.7`). Diese Werte wurden in den Vorab-Versuchen experimentell ermittelt und lieferten für die meisten Datensätze ein einigermaßen günstiges Verhältnis zwischen Clusterzahl und Instanzen pro Cluster.

Mit diesen Parametern erzeugten die Verfahren die in Tabelle 8.2 aufgeführte Anzahl von Clustern. Der SNN-Algorithmus erzeugte sehr häufig Cluster, die aus einem Dokument bestanden, was bei den anderen Verfahren nicht zu beobachten war.

9 Evaluierung

Im Bereich der automatischen Klassifikation (supervised classification) besteht eine Vielzahl an Möglichkeiten, die Güte einer Lösung zu ermitteln. Die Lösung eines Verfahrens ist dann gut, wenn es eine existierende Klasseneinteilung gut nachbildet. Dies ist beim Clustering meist nicht möglich, da eine optimale Klasseneinteilung meist unbekannt ist. Wie kann man aber bestimmen, was eine „gute“ Clustering-Lösung ausmacht? In diesem Kapitel werden Probleme und Möglichkeiten zur Evaluation von Clustering-Lösungen vorgestellt und die Ergebnisse der Experimente aus Kapitel 8.4 anhand geeigneter Kriterien evaluiert.

9.1 Cluster-Validation und mögliche Bewertungskriterien

Die Evaluierung von Clustering-Lösungen wird in der Literatur häufig als „Cluster Validation“ bezeichnet (Kumar 2003, 329). Ziel der Analyse der „Cluster Validity“, die bei einem Clustering-Prozesses durchgeführt werden soll (vgl. Kapitel 2.3), ist die Bestimmung, ob die durch eine Clusteranalyse ermittelte (Struktur-)Beschreibung für die Daten passend ist oder ob sie ein pures Zufallsprodukt darstellt (vgl. Jain et al. 1999, 267 f.). Dies hat den Hintergrund, dass Clustering-Algorithmen fast immer versuchen, irgendwelche Cluster zu erzeugen, sogar dann, wenn die Ausgangsdaten aus zufällig verteilten Datenpunkten bestehen (vgl. Kumar 2003, 331). Eine Aussage über die Güte der berechneten Lösung ist somit für den Anwender sehr hilfreich.

Die Kriterien zur Bewertung können objektiver und subjektiver Natur sein: Von **objektiven** Bewertungskriterien spricht man, wenn strukturelle Eigenschaften der Lösung von Interesse sind, beispielsweise wie gut die einzelnen Cluster voneinander getrennt sind. Als **subjektiv** werden Bewertungskriterien bezeichnet, die den Informationsbedarf des Nutzers berücksichtigen.

Jain und Dubes (1988, 161) nennen drei Arten von Kriterien, mit denen Clustering-Lösungen bewertet werden können:

- Bei den **externen Kriterien** wird die erhaltene Clustering-Lösung mit einer a priori ermittelten Struktur verglichen. Dabei wird beispielsweise die ermittelte Klassenzugehörigkeit einer Instanz mit einer zuvor bestimmten (idealen) Klassenzugehörigkeit verglichen.

- Mit Hilfe von **internen Kriterien** soll ohne Rückgriff auf externe Informationen beurteilt werden, wie passend eine ermittelte Lösung ist. Beispielsweise kann die Fehlerquadratsumme als Gütemaß herangezogen werden.
- Mit **relativen Kriterien** sollen zwei Ergebnisse verglichen werden, um eine Aussage zu treffen, welche der beiden die „bessere“ Lösung ist. Häufig werden dazu interne oder externe Kriterien herangezogen (vgl. Kumar 2003, 333).

9.1.1 Objektive externe Bewertungskriterien

Dieses Kapitel beschreibt Bewertungskriterien, die mittels einer bestehenden (idealen) Clustering-Lösung berechnet werden, ohne dem Informationsbedarf eines Nutzers Rechnung zu tragen.

9.1.1.1 F-Maß

Liegt für die Ausgangsdaten eine z.B. manuell erstellte Einteilung in Cluster vor, so können verschiedene Clustering-Lösungen mit dieser „Ideal-Lösung“ verglichen werden, um z.B. die Auswirkung von Änderung an verschiedenen Parametern bestimmen zu können. Auf dieser Basis („ground truth“, Kumar 2003, 339) kann das F-Maß berechnet werden, das ursprünglich von van Rijsbergen im Kontext der Evaluation von Information Retrieval Verfahren vorgeschlagen und von Larsen und Aone (1999, 18) auf die Evaluation von Dokument-Clustering Verfahren übertragen wurde. Bei der Berechnung wird angenommen, dass jeder Cluster die Antwort auf eine Anfrage (Query) ist und jede Klasse die relevante Menge an Dokumenten darstellt (im Folgenden Notation nach Stein et al. 2003, 217):

Sei D eine Kollektion von Dokumenten und sei $C = C_1, \ldots, C_k$ eine Clustering-Lösung für D. Die von Menschen erstellte Einteilung in Klassen (Referenzklassifikation) sei $C^* = C_1^*, \ldots, C_k^*$. Der Recall wird definiert durch das Verhältnis von der Anzahl der Instanzen, die in Cluster j zur Klasse i gehören, geteilt durch die Gesamtzahl der zur Klasse i gehörenden Dokumente; formal:

$$rec(i, j) = |C_j \cap C_i^*| / |C_i^*|$$

Die Precision von Cluster j hinsichtlich Klasse i errechnet sich aus der Anzahl der Instanzen, die in Cluster j zur Klasse i gehören, geteilt durch die Gesamtzahl der Instanzen in Cluster j, formal:

$$prec(i, j) = |C_j \cap C_i^*| / |C_j^*|$$

Im F-Maß werden Recall und Precision kombiniert zu:

$$F_{i,j} = \frac{2}{\frac{1}{prec(i,j)} + \frac{1}{rec(i,j)}},$$

woraus sich im Folgenden das F-Maß errechnet, das die Güte einer ganzen Clustering-Lösung ausdrückt. Bei vollkommener Übereinstimmung mit der von Menschenhand erstellten Klassifikation ergibt sich für das F-Maß ein Wert von 1:

$$F = \sum_{i=1}^{l} \frac{|C_i^*|}{|D|} * \max_{j=1,\ldots,k} F_{i,j}$$

9.1.1.2 Entropy und Purity

Das Maß der **Entropy** (Entropie) berücksichtigt die Verteilung der Klassen in den einzelnen Clustern einer Lösung. Eine optimale Lösung enthält innerhalb eines Clusters ausschließlich Dokumente einer Klasse, was durch einen Wert von 0 gekennzeichnet wird. Je kleiner der Wert für die Entropie ist, desto besser ist die Clustering-Lösung. Für einen Cluster S_r der Größe n_r gilt (vgl. Zhao und Karypis 2001, 10):

$$E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{q} = \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r},$$

wobei q der Zahl aller Klassen entspricht und n_r^i der Anzahl der Dokumente in Cluster r entspricht, die zur Klasse i gehören. Die (Gesamt-)Entropie ist die nach Clustergröße gewichtete Summe der Entropie-Werte pro Cluster:

$$Entropy = \sum_{r=1}^{k} \frac{n_r}{n} E(S_r)$$

Das Maß der **Purity** „measures the extend [sic!] to which each cluster contained documents from primarily one class." Je größer der Wert für die Purity ist, desto besser ist die Clustering-Lösung. Formal (Zhao und Karypis 2001, 10):

$$P(S_r) = \frac{1}{n_r} \max_i(n_r^i),$$

wobei dies der Anzahl der stärksten Klassengruppe innerhalb eines Clusters entspricht. Die (Gesamt-)Purity errechnet sich durch ein gewichtetes Aufsummieren der Einzelwerte jedes Clusters:

$$Purity = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r)$$

Das Programmpaket CLUTO ermöglicht eine Berechnung der objektiven externen Bewertungsmaße Entropy und Purity, wenn eine Klasseneinteilung für die Clustering-Lösung bekannt ist. Da dies bei den Experimenten im Rahmen dieser Arbeit nicht der Fall ist, können die Maße Entropy und Purity nicht zur Bewertung von Clustering-Lösungen eingesetzt werden.

9.1.2 Objektive interne Bewertungskriterien

Interne Bewertungskriterien können einerseits dazu eingesetzt werden, die intra-Cluster Ähnlichkeit zu beschreiben, d.h. wie dicht liegen die Objekte innerhalb eines Clusters beisammen („cluster cohesion“). Andererseits können sie die inter-Cluster Ähnlichkeit beschreiben, die möglichst gering sein sollte, so dass Cluster gut separiert voneinander sind („cluster separation“ / „cluster isolation“) (vgl. Kumar 2003, 337).

Beim Programmpaket CLUTO werden standardmäßig objektive interne Maße für jeden Cluster und die Gesamtlösung ermittelt. Diese Maße stellen die Ergebnisse der zum Erzeugen der Lösung verwendeten Gütefunktion (z.B. $\mathcal{I}_2$) dar (vgl. Karypis 2003, 19 f.).

Da die anderen Programme, die zum Erzeugen von Clustering-Lösungen im Zuge dieser Arbeit eingesetzt werden, die Ausgabe eines solchen objektiven, internen Maßes von sich aus nicht unterstützen, kann zur Beurteilung der Güte einer Clustering-Lösung auf dieses Maß als Vergleichskriterium nicht zurückgegriffen werden.

9.1.2.1 „Cluster cohesion“

Als Beispiel für ein Maß der „cluster cohesion“ führt Kumar (2003, 338) die Fehlerquadratsumme („sum of squared errors“, SSE) an, die als Gütekriterium beim K-Means Algorithmus eingesetzt wird (vgl. Kapitel 7.2.2). Je kleiner die Streuung innerhalb eines Clusters ist, desto näher liegen die Objekte beieinander.

9.1.2.2 „cluster isolation“

Zur Berechnung der „cluster isolation“ kann man auf die „between sum of squares“ (SSB) zurückgreifen (Kumar 2003, 338), die wie folgt definiert ist:

$$SSB = \sum_{i=1}^{K} |C_i| \sum_{j=1}^{n} (m_j^i - m_j)^2$$

Durch Aufsummieren über alle Cluster erhält man die SSB, wobei m_j^i der j-ten Komponente des i-ten Mittelwerts entspricht, während m_j die j-te Komponente des Gesamtmittelwertes ist. Je höher der Wert der SSB ist, desto besser sind die Cluster voneinander isoliert.

9.1.2.3 Weitere interne Bewertungskriterien

Stein et al. (2003, 217 f.) zählen weitere gängige Bewertungskriterien auf: Beim *Dunn Index* werden ein inter-Cluster Distanzmaß und der Durchmesser eines Clusters zueinander in Beziehung gesetzt. Beim *Davies-Bouldine Index* wird die Streuung innerhalb eines Clusters zur inter-Cluster Distanz in Beziehung gesetzt. Stein et al. weisen darauf hin, dass beiden Maßzahlen eine geometrische Sichtweise von Clustering-Ergebnissen haben und daher nur gut arbeiten, wenn die Cluster eine sphärische Form besitzen. Daher schlagen sie zwei weitere Maße vor (Λ-Maß und $\overline{\rho}$-Maß), die einen graphentheoretischen Ansatz zur Bewertung verfolgen. Im Rahmen ihrer Untersuchung stellten sie die Überlegenheit ihres neuen Maßes $\overline{\rho}$ (expected edge density) fest.

9.1.3 Zusammenfassung der Methoden zur Ermittlung der Cluster Validity

Soll die Güte von Clustering-Lösungen mittels Kennzahlen ausgedrückt werden, bieten sich die in den vorigen Kapiteln beschriebenen Möglichkeiten an. Die Berechnung dieser Kennzahlen ist an bestimmte Voraussetzungen geknüpft. Zum einen muss die Software, die die Clustering-Lösungen erzeugt, diese Kennzahlen von sich aus ermitteln oder über Schnittstellen zur Berechnung der Werte verfügen. Zum anderen muss eventuell eine „Ideal-Lösung“ vorhanden sein, die z.B. als Grundlage für die externen Bewertungskriterien benötigt wird. Da beide Voraussetzungen für die Experimente, die im Zusammenhang mit dieser Arbeit gemacht werden, nicht erfüllt sind, muss nach anderen Möglichkeiten zur Bewertung von Clustering-Lösungen gesucht werden, was im nächsten Kapitel beschrieben wird.

9.2 „Cluster usability“ als subjektives Bewertungskriterium

In dieser Arbeit soll, auf Grund der im vorhergehenden Kapitel genannten Schwierigkeiten, die Güte einer Clustering-Lösung aus Nutzersicht bewertet werden, wozu eine Art von Relevanzbewertung durch Juroren erforderlich ist. Stein et al. (2003, 216) schlagen vor, für diese Art der Evaluierung mittels subjektiver Bewertungskriterien den Begriff *„cluster usability“* zu verwenden.

Relevanzurteile werden bei der Evaluierung von IR-System, z.B. im Rahmen von TREC, zur Ermittlung der Güte eines Systems eingesetzt. Belew nennt mögliche Kritikpunkte an dieser Herangehensweise, die ebenfalls hier in dieser Arbeit einen Einfluss auf das Ergebnis ausüben können (Belew 2000, 116 ff.):

- Die Relevanzbewertungen finden in einer geschützten „Labor-Umgebung“ statt, die dem wirklichen Anwendungsbereich nachempfunden ist. Sie können daher nur als „praxisnah“ gelten, nicht jedoch als „aus der Praxis“ stammend.
- Es stellt sich die Frage nach der Verlässlichkeit von Relevanzbewertungen durch einen einzelnen Juror (intersubject reliability): Die Art und Weise, wie ein Juror entscheidet, hängt von vielen Faktoren ab, so z.B. von den Fachkenntnissen, der verfügbaren Zeit zum Bearbeiten der Aufgaben oder vom Stil, in dem die Dokumente abgefasst sind.
- Außerdem stellt die Art der Ergebnispräsentation einen weiteren Einflussfaktor auf das Verhalten eines Jurors dar: Bekommt er das ganze Dokument zum Beurteilen oder werden ihm nur Teile daraus präsentiert?

Der Kritikpunkt der „intersubject reliability“ wurde indirekt in einem Gespräch mit einem Juror bestätigt. Er meinte, dass er einen Tag später seine getätigten Bewertungen, von seinem derzeitigen Standpunkt aus, nicht mehr derart vergeben hätte. Außerdem, so derselbe Juror, habe er beim erneuten Bewerten der gleichen Anfrage (auf Grund eines eingetretenen Datenverlusts) teilweise anders bewertet, als er es zuvor getan hatte.

9.2.1 Methodik

Die Juroren (12 Studenten, Laien im Umgang mit Patentdokumenten und ohne genaue Kenntnis der Arbeitsweise der eingesetzten Clustering-Verfahren) erhielten mehrere Clustering-Lösungen präsentiert und entschieden für jedes Dokument eines Clusters, ob dieses Dokument in den Gesamtkontext des jeweiligen Clusters passt oder nicht hinein passt.

Zur Erfassung der Relevanzurteile wurde eine im Zuge der Magisterarbeit in JAVA entwickelte Anwendung erstellt (ClustEv = **Clus**tering-Lösungs **Ev**aluations-Tool), um die Juror-Urteile zu speichern und automatisch auszuwerten. Eine Beschreibung der Fähigkeiten und Eigenschaften dieses Programms befindet sich im Anhang in Kapitel B.3.

Jeder Juror erhielt ein Manual, in dem die Installation und Handhabung des Tools ClustEv beschrieben wurde, einen Arbeitsauftrag und einen Fragebogen. Pro Anfrage wurde zur Motivation der Juroren im Arbeitsauftrag ein Informationswunsch formuliert. Dies soll dazu dienen, die Tätigkeit der Juroren in einen gewissen Rahmen

einzubetten. Beispielsweise lautete der Arbeitsauftrag für die Anfrage „bild? (S) verarbeit? AND G06F017/ICM?“: „Du möchtest einen Überblick über Patente aus dem Bereich Bildverarbeitung oder Verarbeitung von Bildern bekommen.“

Bei der Bewertung erhielten die Juroren die Vorgabe, dass pro Cluster nur ein Grobkonzept vorhanden sein sollte. Sind innerhalb eines Clusters mehrere Themenbereiche zusammen gruppiert, so musste sich der Juror für ein vorherrschendes Themengebiet entscheiden und die anderen Dokumente mit „nicht passend“ bewerten. Gehören die in einem Cluster gruppierten Konzepte zu einem umfassenderen Konzept, das sich beispielsweise intellektuell aus dem Zusammenhang der Dokumente erschließen lässt, so sind die zugehörigen Dokumente mit „passend“ zu bewerten. Die Auswertung mit dem Tool ClustEv liefert Angaben zum:

- Bewertungsverhalten eines einzelnen Jurors: Wie bewertete er/sie die Cluster einer Anfrage? Wie viele Instanzen wurden mit „passend“ oder „nicht passend“ bewertet? Wurde eine Anfrage auf Basis der aufsummierten Einzelstimmen insgesamt als „passend“, „nicht passend“ oder bei unvollständiger Bewertung aller Dokumente als „nicht vollständig“ bewertet?
- Bewertungsverhalten der Juroren bezüglich einer Anfrage. Dazu werden die Einzelurteile der Juroren pro Anfrage zusammengezählt. Für eine Anfrage werden sowohl die Gruppen-Bewertung (alle Juroren einer Anfragegruppe) pro Cluster ermittelt, als auch die aufsummierten Absolutwerte der Urteile („passend“, „nicht passend“ oder „nicht vollständig“) zur Bildung einer Gesamtbewertung einer Anfrage.

Zusätzlich sollten die Juroren auf einem Papier-Fragebogen angeben,

- wie der Gesamteindruck der Clustering-Lösung pro Anfrage auf sie wirkt (Schulnoten-Skala von 1 = sehr gut bis 6 = mangelhaft).
- ob die Anzahl der erzeugten Cluster für die jeweilige Anfrage passend war, oder nicht. Falls mit „nein“ geantwortet wurde, wurde gefragt, ob zu viele oder zu wenige Cluster erzeugt wurden.
- ob zu bestimmten Anfragen oder im Allgemeinen etwas auffiel (Kommentare).

9.2.2 Erhebungsplan

Zur Durchführung der Experimente wurden die in Kapitel 8.1.2.2 ausgewählten acht Anfragen herangezogen und mit den drei Clustering-Verfahren („bisecting K-Means“, SNN und einem probabilistischen Verfahren) verarbeitet. Um die Annahme zu überprüfen, dass Patentfamilien-Doppel die erzeugten Clustering-Lösungen verzerren könnten, werden den Juroren sämtliche Ergebnisse sowohl mit, als auch ohne Patentfamilien-Doppel vorgelegt. Da die Bewertung sehr zeitaufwändig ist, erhielt jeder Juror/jede Jurorin nur einen Teil der Anfragen (siehe Tabelle 9.1). Die Juroren der

Gruppe	Juroren	Anfragen
A	Juroren A1–A3	bild? (S) verarbeit? AND G06F017/ICM medizin? AND G06F017/ICM bild_verarbeit_ipc_mD medizin_ipc_md
B	Juroren B1–B3	datenuebertragung? AND G06F017/ICM server? AND client? AND G06F017/ICM digital? AND bild? AND G06F017/ICM
C	Juroren C1–C3	brows? AND G06F017/ICM multimedia? AND G06F017/ICM navig? AND G06F017/ICM

Tabelle 9.1: Aufteilung der Anfragen auf die Juroren

Gruppe B und C erhielten jeweils 3 Anfragen (mit/ohne PF-D) während Gruppe A nur 2 Anfragen (mit/ohne PF-D) erhielt. Den Juroren der Gruppe A wurde zusätzlich eine „Pseudo-Lösung“ zum Bewerten präsentiert (im Rahmen der Bewertung von Anfragen des SNN-Algorithmus), deren Cluster-Einteilung der Einteilung in Untergruppen der IPC entspricht (Anfragen medizin_ipc_mD und bild_verarbeit_ipc_mD, jeweils mit PF-D). Dieses Vorgehen soll einen Vergleich ermöglichen, wie gut die Nutzer eine alleinige Gruppierung anhand der IPC bewerten.

9.3 Auswertung der Experimente

Die auf zwölf Juroren aufgeteilten Anfragen (siehe Tabelle 9.1) wurden nicht alle vollständig bewertet. Zwei Drittel der Juroren bearbeitete die ausgegebenen Anfragen vollständig (d.h. in jeder Gruppe waren dies mindestens zwei Juroren), das restliche Drittel führte die Bewertung in unterschiedlichem Umfang durch. Gründe für die unvollständige Bewertung wurden von den Juroren auf den Papierfragebögen und in Gesprächen genannt, auf die in Kapitel 9.3.3 näher eingegangen wird.

9.3.1 Auswertung der Juroren-Beurteilungen auf Dokumentebene

Die Juroren sollten für jedes Dokument entscheiden, ob es in den Gesamtkontext des Clusters passt oder nicht hinein passt. Die Einzelbewertungen eines Jurors wurden innerhalb der Anfragegruppe (A, B, oder C) mittels des Tools ClustEv aufsummiert. Um eine Aussage bezüglich eines bestimmten Clustering-Verfahrens (bisecting K-Means, SNN oder des probabilistischen Verfahrens) zu erhalten, wurden pro Verfahren die zuvor berechneten Werte über alle Anfragegruppen aufsummiert. Hierbei wurde nach Anfragen unterschieden, in denen PF-D vorkamen *(md)* bzw. nicht vorhanden waren *(od)*. Außerdem wurde die Gesamtsumme der Bewertungen einer Anfrage (zusammengesetzt aus *md + od*) berechnet, was in Tabelle 9.2 und den Abbildungen 9.1 und 9.2 dargestellt ist.

	passend	passend (%)	passend / Clusteranz. (‰)	nicht passend	nicht passend (%)	nicht passend/ Clusteranz. (‰)	nicht bewertet	Gesamtzahl bew. Dok.	Gesamtzahl Dok.
probabilist. Verf. md	1098	39	4,22	1700	61	6,53	1174	2798	3972
probabilist. Verf. od	863	39	4,49	1371	61	7,14	1042	2234	3276
probabilist. Verf. gesamt	1961	39	2,18	3071	61	3,41	2216	5032	7248
bisecting K-Means md	1841	51	5,09	1775	49	4,91	356	3616	3972
bisecting K-Means od	1230	46	5,65	1456	54	6,69	590	2686	3276
bisecting K-Means gesamt	3071	49	2,69	3231	51	2,83	946	6302	7248
SNN md	1751	64	1,83	979	36	1,02	1116	2730	3846
SNN od	1254	56	2,08	999	44	1,66	923	2253	3176
SNN gesamt	3005	60	0,98	1978	40	0,64	2039	4983	7022

Tabelle 9.2: Bewertungen der Juroren auf Dokumentebene

Zunächst wird auf die Absolutwerte der Juror-Urteile („passend“ / „nicht passend“) eingegangen. Dies geschieht, indem das Verhältnis zwischen der Gesamtzahl der bewerteten Dokumente und der Anzahl der mit „passend“ bzw. „nicht passend“ bewerteten Dokumente berechnet wird. Dabei wird Folgendes festgestellt:

- Der SNN-Algorithmus weist die meisten mit „passend“ und die wenigsten mit „nicht passend“ bewerteten Dokumente auf. Sowohl die Datensätze mit PF-D, als auch ohne, wurden eindeutig mit „passend“ bewertet.
  Einschränkend muss jedoch hinzugefügt werden, dass dieses Verfahren eine sehr große Zahl an ein-elementigen Clustern erzeugt, die die Juroren überwiegend mit „passend“ beurteilten. Diese große Anzahl an „passenden“ Clustern verschiebt das Ergebnis zu Gunsten des SNN-Algorithmus. Daher wird im Weiteren zusätzlich eine Normierung anhand der Anzahl der erzeugten Cluster durchgeführt und die Ergebnisse anhand dieser Datengrundlage nochmals beurteilt.
- Das „bisecting K-Means“-Verfahren kann anhand der Juror-Urteile nicht eindeutig als „passend“ oder „nicht passend“ beurteilt werden. Bei den Datensätzen mit PF-D beträgt der Unterschied zwischen „passend“ und „nicht passend“ gerade einmal 2 %, was nicht sonderlich aussagekräftig ist. Die Datensätze ohne PF-D wurden mit 8 % Unterschied eher als „nicht passend“ bewertet. Diese kleinen Unterschiede führen in der Gesamtbewertung zu einer Patt-Situation ohne eine eindeutige Tendenzaussage, da zwischen „passend“ und „nicht passend“ ein minimaler Unterschied von 2 % besteht. Dies ist bei den beiden anderen Verfahren nicht der Fall.

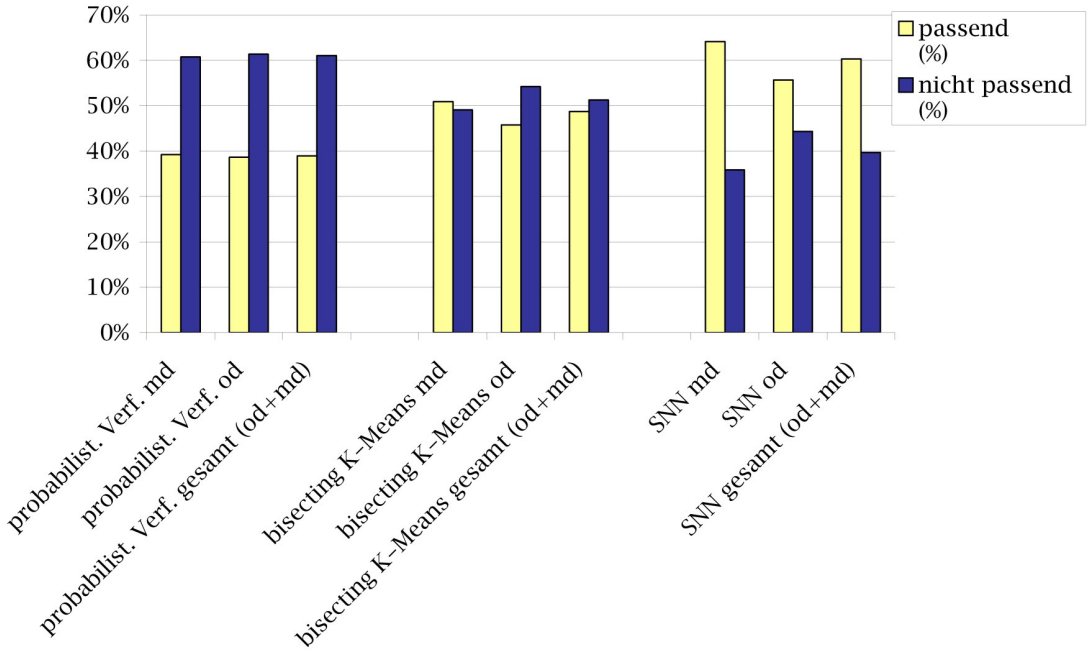


Abbildung 9.1: Bewertungen der Juroren auf Dokumentebene (Absolutwerte)

- Bei dem probabilistischen Verfahren, das mittels Autoclass-C überprüft wurde, stellt sich in allen Datensätzen (mit/ohne PF-D) heraus, dass die Juroren die erzeugten Lösungen überwiegend mit „nicht passend“ bewerteten.

Auf Grund der Eigenschaft des SNN-Algorithmus, viele ein-elementige Cluster zu erzeugen, wurde versucht, die vorliegenden Daten vergleichbarer zu machen. Daher wurde eine Normalisierung anhand der Clusteranzahl durchgeführt, um den Effekt der zahlreichen ein-elementigen SNN-Cluster zu kompensieren. Das der Auswertung zu Grunde liegende Verhältnis errechnet sich wie zuvor beschrieben, wird jedoch durch die Gesamtzahl der erzeugten Cluster für eine bestimmte Datensatzgruppe (md, od oder gesamt) geteilt. Auf Basis dieser „normierten“ Berechnung (siehe Abbildung 9.2) leiten sich folgende Beobachtungen ab:

- Nach der „Normierung“ verfügt das „bisecting K-Means“-Verfahren, im Vergleich zu dem probabilistischen Verfahren und dem SNN-Algorithmus, über den größten Anteil an Dokumenten, die mit „passend“ bewertet wurden (in allen Datensätzen, egal ob mit oder ohne PF-D). Bei den Datensätzen ohne PF-D ist der Anteil der mit „nicht passend“ bewerteten Dokumente größer, als der der Datensätze mit PF-D bzw. dem Gesamtwert (md + od).
- Bei dem probabilistischen Verfahren überwiegt in allen drei Datensatzvarianten (md, od, gesamt) die Bewertung mit „nicht passend“. Der Anteil der mit „passend“ beurteilten Dokumente ist bei den durch das „bisecting K-Means“-Verfahren berechneten Ergebnissen größer.

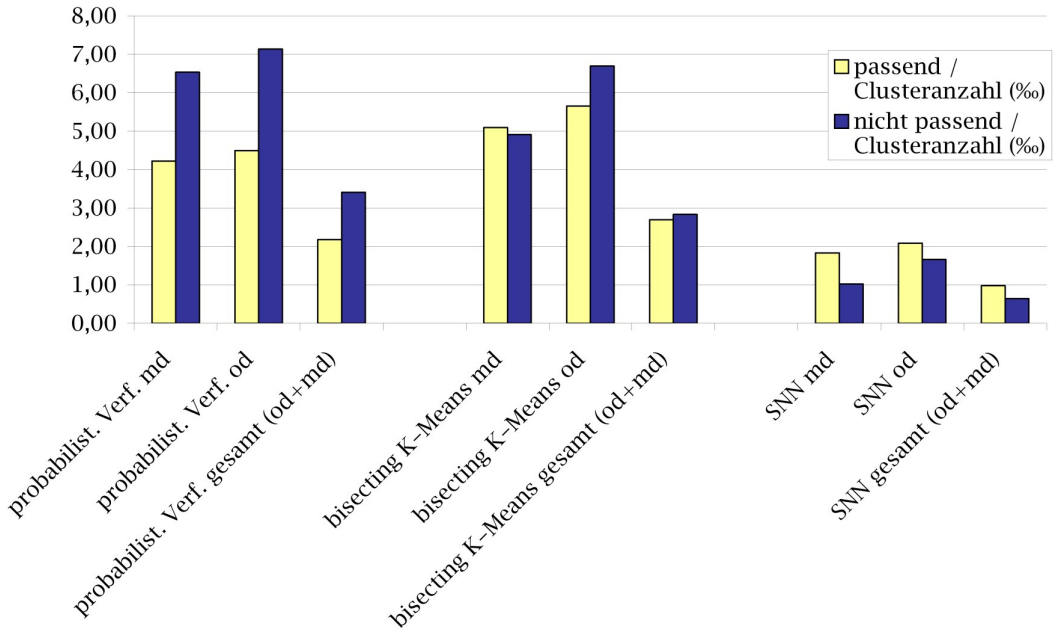


Abbildung 9.2: Bewertungen der Juroren auf Dokumentebene (Normiert anhand der Anzahl erzeugter Cluster)

- Der SNN-Algorithmus schneidet nach der „Normierung“ sehr schlecht ab. Jedoch überwiegt in allen drei Datensatz-Varianten (md, od, gesamt) die Bewertung mit „passend“.
- Bei allen drei Verfahren ergibt sich das Bild, dass mehr Dokumente mit „passend“ bewertet werden, wenn die Ausgangsdaten ohne PF-D gewählt wurden.

9.3.2 Auswertung nach Vergabe von Schulnoten durch die Juroren

Die Juroren wurden gebeten, auf Papier-Fragebögen jede zu bearbeitende Anfrage mittels Vergabe von Schulnoten zu beurteilen. Dabei gilt, dass ein optimales Verfahren die Note 1, ein sehr schlechtes Verfahren die Note 6 erhalten soll. Um zu einer Gesamtbewertung für die drei eingesetzten Clustering-Verfahren zu gelangen, wurden die Benotungen getrennt nach Art der Datensätze (mit bzw. ohne PF-D) aufsummiert und durch die Anzahl der Juroren geteilt, die für diese Datensatzart eine Bewertung getätigt haben. Die Ergebnisse sind je nach Anfrage und Gruppe (Tabelle 9.4), sowie über alle Verfahren (siehe Tabelle 9.3 und Abbildung 9.3) dargestellt. Aus den Ergebnissen lässt sich Folgendes ablesen:

- Vergleicht man die Verfahren anhand der Gesamtwerte (md + od), so ergibt sich die Reihenfolge „bisecting K-Means“, SNN-Algorithmus und probabilistisches Verfahren.
- Beim „bisecting K-Means“-Verfahren liegen die Nutzerbewertungen für alle drei Berechnungsarten (md, od, gesamt) dicht beieinander.

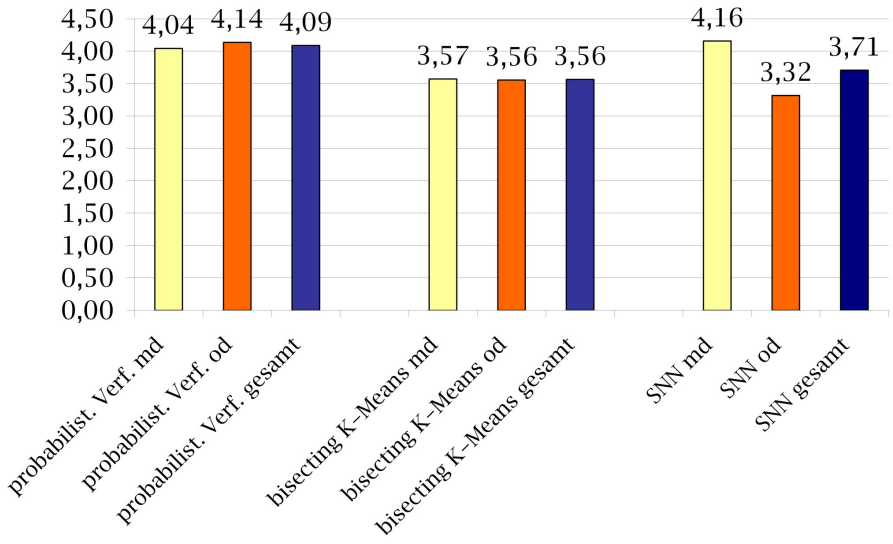


Abbildung 9.3: Bewertung nach Schulnoten

Verfahren	Abgegebene Stimmen	Summe Einzelnoten	Gesamtnote
probabilistisches Verfahren md	23	93	4,04
probabilistisches Verfahren od	22	91	4,14
probabilistisches Verfahren gesamt	45	184	4,09
„bisecting K-Means“ md	28	100	3,57
„bisecting K-Means“ od	27	96	3,56
„bisecting K-Means“ gesamt	55	196	3,56
SNN md	19	79	4,16
SNN od	22	73	3,32
SNN gesamt	41	152	3,71
Pseudo-Lösung (nach IPC): bild_verarbeit	4	13	3,00
Pseudo-Lösung (nach IPC): medizin	4	12	3,25
Pseudo-Lösung (nach IPC): gesamt	8	25	3,13

Tabelle 9.3: Bewertungen nach Schulnoten (Pseudo-Lösung wurde nur von vier Juroren bewertet.)

- Die Juroren bevorzugten beim SNN-Algorithmus die Datensätze, die ohne PF-D zusammengestellt wurden.
- Innerhalb der Juroren-Gruppe A wurde das bisecting K-Means-Verfahren mit der Note 2,73 (gesamt) am besten beurteilt. Die Pseudo-Lösung (Note 3,13), die die Einteilung in Untergruppen der IPC widerspiegelt, landete nach SNN (Note 2,67) auf dem dritten Platz (bezogen auf Gesamt-Werte).

Insgesamt wurde keines der getesteten Verfahren in der Gesamtbetrachtung (md + od) als herausragend gut (z.B. durch Note 2 und besser) bewertet. Vielmehr spielte sich die Bewertung (über die verschiedenen Betrachtungsarten, d.h. md, od oder gesamt) in einem Bereich zwischen den Noten 3,56 und 4,16 ab, der mit befriedigend bis ausreichend bezeichnet werden kann.

		„bisecting K-Means“						SNN						probabilistisch				
Gruppe	Anfrage	Juror 1	Juror 2	Juror 3	Juror 4	Note		Juror 1	Juror 2	Juror 3	Juror 4	Note		Juror 1	Juror 2	Juror 3	Juror 4	Note
	bild_verarbeit_md	3	3	2	4	3,00		4	3	5		4,00		4	4	3		3,67
	bild_verarbeit_od	3	5	2	4	3,50		3	3	2		2,67		4	5	1		3,33
	bild_verarbeit_ipc_md							3	3	2	5	3,25						
	medizin_ipc_md							3	3	2	4	3,00						
A	medizin_md	2	2	2		2,00		4	2	3		3,00		4	4	2	4	3,50
	medizin_od	2	2	2	3	2,25		3	2	3		2,67		4	5	2	3	3,50
	md			7*	18**	2,57				6*	21**	3,50				7*	25**	3,57
	od			8	23	2,88				6	16	2,67				7	24	3,43
	gesamt (ohne _ipc)			15	41	2,73				12	37	3,08				14	49	3,50
	digital_bild_md	4	4	4		4,00		4	2			3,00		5	3		3	3,67
	digital_bild_od	4	3	4		3,67		4	2			3,00		5	2			3,50
	datenuebertragung_md	3	5	2	5	3,75		4	2		6	4,00		5	3			4,00
	datenuebertragung_od	3	3	3		3,00		4	2		4	3,33		5	4			4,50
B	server_client_md	5	2	3		3,33		5	2			3,50		6	3			4,50
	server_client_od	5	4	2		3,67		5	2			3,50		6	3			4,50
	md			10	37	3,70				7	25	3,57				7	28	4,00
	od			9	31	3,44				7	23	3,29				6	25	4,17
	gesamt			19	68	3,58				14	48	3,43				13	53	4,08
	brows_md	5	4	3	5	4,25		6	6	1		4,33		6	5	3		4,67
	brows_od	5	5	4	4	4,50		6	5	2		4,33		6	6	5		5,67
	multimedia_md	3	5	3	6	4,25		5	5	1		3,67		4	5	3		4,00
	multimedia_od	4	5	3		4,00		5	5	1		3,67		4	6	2		4,00
C	navig_md	5	4	2		3,67		3	5	1		3,00		5	6	3		4,67
	navig_od	5	5	2		4,00		4	5	1		3,33		5	6	2		4,33
	md			11	45	4,09				6	33	5,50				9	40	4,44
	od			10	42	4,20				9	34	3,78				9	42	4,67
	gesamt			21	87	4,14				15	67	4,47				18	82	4,56

Tabelle 9.4: Bewertungen nach Schulnoten für alle Anfragen und Gruppen
(* = Anz. abgegebene Stimmen, ** = Summe der abgegebenen Noten)

9.3.3 Auswertung der Juroren-Kommentare auf den Papier-Fragebögen

Die Schwierigkeiten der Juroren mit der Art von Aufgabenstellung und den Patentdokumenten an sich spiegelt sich in den Kommentaren auf den Papier-Fragebögen wider. So wird von ihnen fast ausnahmslos eine schwere Verständlichkeit der Patentdokumente beschrieben. Zurückgeführt wird dies zum einen auf die Sprache, in der die Patentdokumente abgefasst wurden („lange Sätze, die sich über mehr als 20 Zeilen erstrecken.“ [Juror A1], „Patentinhalte konnten eher schlecht verstanden werden ohne Abstract.“ [Juror A4], „Patentdokumente sind schwer verständlich und u.U. sehr speziell, so daß es für mich schwer war, Dokumente zu Konzepten zuzuordnen.“ [Juror B1]). Zum anderen hängt die schwere Verständlichkeit – so die Juroren-Meinungen – von der gewählten Thematik der Anfragen ab. Viele Juroren beschreiben, dass sie nicht kompetent genug wären, die häufig in einer sehr speziellen Fachsprache beschrieben Sachverhalte nachzuvollziehen („Fehlende Fachkompetenz.“ [Juror B3]) und auf Grund mangelnden Verständnisses eine genaue Bewertung nicht möglich wäre.

Zudem wurde der Wunsch nach einer Aufteilung der großen Cluster in mehrere kleinere Cluster mehrfach genannt. Die Mehrfachnennung dieses Punktes in Verbindung mit dem Wissen, dass die Art und Weise der Clustererzeugung den Juroren

nicht bekannt war, lässt darauf schließen, dass die Nutzer eine gewisse Kontrolle über das Clustering-Ergebnis ausüben wollen. Das entspricht der von Fattori et al. (2003, 336) angeführten Beobachtung, dass Nutzer keine „Black-Box“-Werkzeuge verwenden wollen und steuernd eingreifen wollen, siehe Kapitel 4.4.2.

In den mündlich geführten Gesprächen über den Verlauf und Fortschritt der Bewertung wurde ausnahmslos von allen Juroren die Bearbeitung als sehr zeitaufwändig und anstrengend beschrieben. Von den Juroren wurde dabei auch mehrfach genannt, dass sie Schwierigkeiten haben, ein gemeinsames Konzept innerhalb eines Clusters zu erkennen. Nachfolgend werden die Anmerkungen der Juroren zu den einzelnen Verfahren wiedergegeben:

probabilistisches Verfahren (Autoclass-C)

- „Dokumente mit gleichem Namen sind nicht in gleichem Cluster“ (Juror A1)
- „Konzept kaum erkennbar.“ (Juror A1)
- „Die Cluster waren sehr zusammengewürfelt, daher oft schwierig ein einziges Konzept für ein Cluster festzulegen.“ (Juror C3)
- „Interessanterweise sind die großen Cluster oft besser zusammengestellt als die kleineren (hinsichtlich des gemeinsamen Konzepts der einzelnen Patente)“. (Juror A1)
- „Etwas unpräzise Ergebnisse.“ (Juror A3)
- „Was Autoclass genau macht, ist mir ein Rätsel.“ (Juror B1)

„bisecting K-Means“ (CLUTO)

- „Hier erscheint Clusterbildung logisch, Konzept gut erkennbar.“ (Juror A1)
- „Cluster sind besser zusammengestellt als bei Autoclass. Daher ist es leichter das gemeinsame Konzept in einem Cluster zu erkennen. Die Cluster sind insgesamt recht gut zusammengestellt, obwohl es auch hier Verbesserungen gäbe; [. . .]“ (Juror C3)

SNN

- Cluster 1 zu groß (mehrfach genannt): „Das Cluster 1 ist mit 25 Elementen zu groß.“ (Juror A2).
- „Zu viele kleine Cluster, die zugeordnet werden müssten.“ (Juror A1); „Die einelementigen Cluster sind nicht sehr aussagekräftig.“ (Juror A2)
- „Die Cluster in diesem Verfahren sind meiner Meinung nach am Besten zusammengestellt.“ (Juror C3)
- „Cluster mittlerer Größe sind in Ordnung.“ (Juror A1)

Bewertung der Pseudo-Lösung, erstellt nach den IPC-Untergruppen

Diese Pseudo-Lösung wurde den Juroren bei der Bewertung der vom SNN-Algorithmus erzeugten Clustering-Lösungen vorgelegt. Insgesamt gesehen beurteilten die Juroren dieses „Verfahren“ nicht ausdrücklich besser oder schlechter als andere Verfahren, was sich auch in der im vorhergehenden Kapitel (9.3.2) dargestellten Auswertung der Schulnotenvergabe widerspiegelt. In ihren Kommentaren formulierten die Juroren folgende Punkte:

- „In den großen Clustern ist tendenziell erkennbar, worum es geht, jedoch zwei Gruppen in einem Cluster.“ (Juror A1)
- „Cluster 2 zu groß, ansonsten besser als vorherige Ergebnisse von SNN.“ (bild_verarb_ipc, Juror A3)
- „Cluster 2 zu vage; Cluster 6 dito“ (medizin_ipc_md, Juror A4)

9.3.4 Bewertung der erzeugten Clusteranzahl

Auf den Papier-Fragebögen der Juroren wurde für jede zu bearbeitende Anfrage gefragt, ob die Anzahl der erzeugten Cluster „passend“ war oder ob „zu viele“ bzw. „zu wenige“ Cluster erzeugt wurden. Die Ergebnisse der Nennungen sind in Tabelle 9.5 und in Abbildung 9.4 dargestellt.

	passend	zu viele	zu wenige
probabilistisches Verfahren md	15	2	6
probabilistisches Verfahren od	14	2	6
probabilistisches Verfahren gesamt	29	4	12
„bisecting K-Means“ md	18	2	7
„bisecting K-Means“ od	17	1	7
„bisecting K-Means“ gesamt	35	3	14
SNN md	3	17	2
SNN od	3	17	2
SNN gesamt	6	34	4
Pseudo-Lösung (nach IPC): gesamt	1	2	5

Tabelle 9.5: Bewertung der erzeugten Clusteranzahl

Die mittels des probabilistischen Verfahrens und des „bisecting K-Means“-Verfahrens erzeugte Clusteranzahl wurde von den Juroren mehrheitlich mit „passend“ bewertet, nur wenigen Juroren war die Clusteranzahl zu groß. Häufiger wünschten sich die Juroren mehr Cluster. Diese Beobachtung ist für alle drei Betrachtungsarten (Datensätze mit PF-D, ohne PF-D und gesamt) einheitlich.

Beim SNN-Algorithmus bewertete die Mehrheit der Juroren die Clusteranzahl mit zu zahlreich. In den Fällen, in denen die Nutzer mehr Cluster wünschten, bezieht sich dies (so die Kommentare in den Papier-Fragebögen) auf den ersten, sehr großen Cluster. Auch diese Beobachtung ist für alle drei Betrachtungsarten (md, od, gesamt) einheitlich.

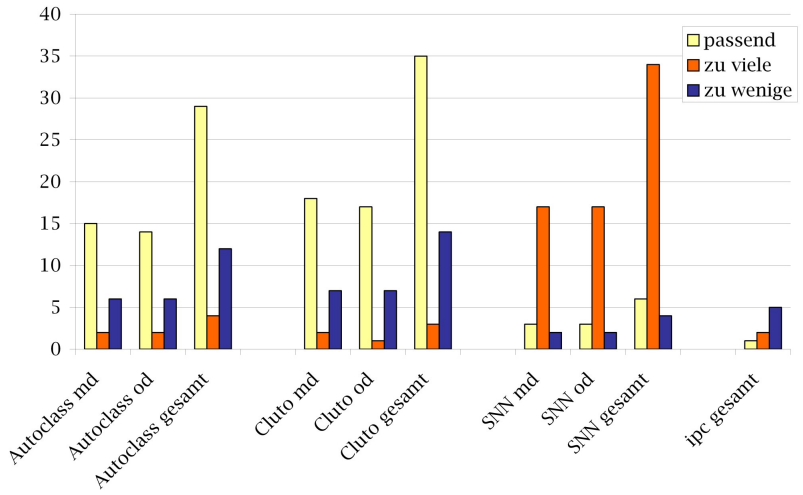


Abbildung 9.4: Bewertung der erzeugten Clusteranzahl

Die Gruppe A, die eine nach den IPC Untergruppen erstellte Pseudo-Lösung zum Bewerten erhielt, vergab überwiegend die Bewertung, dass zu wenige Cluster vorhanden waren.

9.4 Schlussfolgerungen aus den Experimenten

Ziel der Experimente, die in diesem Kapitel ausgewertet wurden, ist die Überprüfung der zu Beginn von Kapitel 8 vorgestellten Annahmen. Als Ergebnis lässt sich nach der Auswertung der Daten Folgendes formulieren.

Annahme 1: *Das Entfernen von PF-D erzeugt eine bessere Clusterqualität.*
Das Vorhandensein bzw. Nicht-Vorhandensein von PF-D in den Ausgangsdaten spielte bei der Bewertung durch die Juroren keine große Rolle. Weder in der Auszählung der Bewertungen auf Dokumentebene, noch auf Basis der Vergabe von Schulnoten durch die Juroren, konnte eine Tendenz für oder gegen das Filtern von PF-D ermittelt werden. Auch in den Kommentaren der Juroren finden sich hinsichtlich dieser Thematik keine Nennungen. Daher kann davon ausgegangen werden, dass die Entfernung von PF-D keinen großen Vorteil zu einer Qualitätsverbesserung von Clustering-Lösungen beiträgt.

Annahme 2: *Ein Verfahren zur Erzeugung von Clustering-Lösungen sticht mit qualitativ hochwertigen Lösungen deutlich hervor.*
Betrachtet man die Schulnoten-Bewertungen der Juroren für die Verfahren, so liegen sie relativ dicht beieinander in einem Spektrum, das mit Noten zwischen 3,56 und 4,16 als befriedigend bis ausreichend beschrieben werden kann. Die Ausgangsdaten (d.h. welche Suchanfragen an eine Patentdatenbank gestellt wurden) spielen eine große Rolle, da z.B. innerhalb der Gruppe A das „bisecting K-Means“-Verfahren

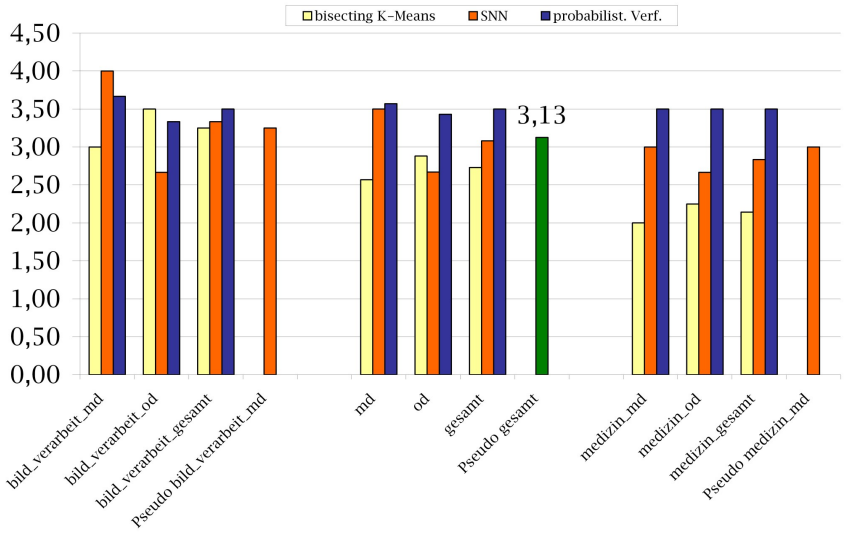


Abbildung 9.5: Bewertung nach Schulnoten - Gruppe A mit Pseudo-Lösung. Der mittlere Block gibt die Summen der Anfragen *bild_verarbeit* und *medizin* an.

häufig mit der Note „gut“ bewertet wurde, wohingegen bei anderen Gruppen mit anderen Anfragen generell schlechtere Noten (drei und schlechter) vergeben wurden. Das legt den Schluss nahe, dass die Auswahl der Anfragen eine größere Wirkung auf das Ergebnis hat, als die getesteten Clustering-Verfahren selbst.

Insgesamt wird das in CLUTO implementierte „bisecting K-Means“-Verfahren sowohl von allen Juroren in der Schulnoten-Bewertung, als auch in der Auszählung auf Dokumentebene (normiert) am besten bewertet, was aber auf Grund der geringfügigen Unterschiede zu den Andersplatzierten höchstens als Tendenzaussage gewertet werden kann.

Annahme 3: *Die Gruppierung von Patentdokumenten mittels der IPC-Klassen ist per se ideal.*
Ohne Kenntnis der Entstehung der Gruppierung bewerteten die Juroren der Gruppe A die Pseudo-Lösung als drittbestes Verfahren nach „bisecting K-Means“ und SNN (Schulnoten gesamt, siehe Abbildung 9.4). In den Kommentaren reichten die Stimmen von “Cluster 2 zu viele [Dokumente], ansonsten besser als vorherige Ergebnisse von SNN.“ (bild_verarb_ipc_md, Juror A3) und „Cluster 7: Kaum Gruppierung? Cluster kaum erkennbar. Cluster 2: viel zu groß.“ (medizin_ipc_md, Juror A4) bis hin zu „Einige Cluster bspw. 15, 13, 9 könnten eventuell zusammengefasst werden.“ (bild_verarb_ipc_md, Juror A2). Insgesamt gesehen fand diese Art der Gruppierung bei den Juroren der Gruppe A keinen großen Anklang (Schulnotenvergabe und Kommentare). Da die Beurteilung der Pseudo-Lösung nur in Gruppe A durchgeführt wurde, ist ein Vergleich mit den Gesamturteilen, die auf einer breiteren Bewertungsbasis entstanden sind, kaum möglich.

10 Fazit und Ausblick

In dieser Arbeit wurden verschiedene Ansätze zum Clustern von Patentdokumenten vorgestellt und mittels eines Experiments auf ihre Eignung angesichts des Anwendungsbereichs Patentrecherche und Patentinformation untersucht. Drei Arten von Algorithmen („bisecting K-Means“ als partitionierendes Verfahren, SNN als ein Verfahren mit alternativer Distanzberechnung sowie ein probabilistisches Verfahren) wurden auf Basis von Nutzerbewertungen miteinander verglichen.

Die Analyse der Clustering-Verfahren in einem bestimmten Anwendungskontext ist deshalb wichtig, da es eine objektiv beurteilbare, allgemein gültige, *„optimale“* Clustering-Lösung nicht gibt. Je nach Anwendungsgebiet kann ein anderes Verfahren zur Cluster-Erzeugung auf Grund subjektiver oder objektiver Kriterien „geeigneter“ sein. Dies wurde für den Anwendungsbereich Patentrecherche und -information hier versucht zu ermitteln.

Bei der Erzeugung der Clustering-Lösungen mittels der untersuchten Clustering-Verfahren spielen nachfolgend genannte Faktoren eine Rolle, Einfluss auf das Endergebnis ausgeübt haben:

- Die Auswahl der Anfragen zur Ermittlung der Datengrundlage.
  - Die Anfragen für die Clustering-Läufe basieren nicht auf realen Anfragen und können somit höchstens als praxisnah, nicht jedoch mit „aus der Praxis“ bezeichnet werden.
  - Manche Anfragen wurden generell besser bewertet als andere Anfragen, unabhängig vom Verfahren, das zur Clusterbildung gewählt worden ist.
  - Die Juroren, die die Anfragen bewerteten, waren Laien im Umgang mit Patentdaten.
- Die Art und Weise der Datenaufbereitung:
  - Elimination von Stoppwörtern (Umfang und Inhalt der Stoppwortliste),
  - der verwendete Stemming-Algorithmus (dessen Mächtigkeit und Qualität, z.B. dessen Fähigkeit zur Kompositazerlegung),
  - das Schema zur Termgewichtung (und eventuell die Wahl der Parameter für das Verfahren)
  - die festgelegte Mindestanzahl an Termen (5 pro Dokument).

In der Auswertung der Evaluations-Ergebnisse konnte gezeigt werden, dass die drei Verfahren von den Nutzern eher skeptisch hinsichtlich ihrer Eignung bewertet wurden (auf einer Schulnotenskala entspricht dies den Noten ausreichend bis befriedigend). Von den vorgestellten Verfahren erzielte das in CLUTO implementierte

„repeated bisecting K-Means“-Verfahren die besten Ergebnisse, wobei dies nur als Tendenzaussage gelten kann, da zu den anderen Verfahren nur relativ geringe Unterschiede bestehen (basierend auf der Benutzerbewertung durch Schulnoten).

Auch praktische Gründe sprechen für eine weitere Untersuchung der partitionierenden Verfahren: Der SNN-Algorithmus erweist sich als sehr schwer zu parameterisieren, so dass kleine Änderungen an den Parametern ein vollkommen anderes Clustering-Ergebnis erzeugen können. Außerdem erhält der Nutzer je nach Parameterkonstellation eine Vielzahl von Clustern, die nur ein Dokument aufweisen, was im Rahmen der Nutzerbewertung als nachteilige Eigenschaft angemerkt worden ist. Gegen das probabilistische Verfahren (implementiert in Autoclass-C) spricht, neben der im Vergleich zu den anderen Verfahren etwas schlechteren Nutzerbewertung, die Tatsache, dass die Laufzeit zur Ermittlung eines Ergebnisses sehr hoch ist und mit steigender Attributzahl extrem anwächst. So dauerte die Berechnung der Clustering-Lösung für die Anfrage „bild? (S) verarbeit?“ mit 100 Dokumenten und 2554 Attributen bei Autoclass-C insgesamt 15 Minuten und 36 Sekunden, wohingegen die Laufzeit bei CLUTO im Millisekundenbereich lag (0,741 ms zur Berechnung, 0,27 ms zum Einlesen der Quelldaten).

Ideen für Anknüpfungspunkte zu weitergehenden Untersuchungen entstanden überwiegend im Rahmen der Durchführung und Vorbereitung der Experimente. Hinsichtlich der Datenbasis ist zu erwägen, ob eher mit Volltexten aus der Datenbank PATDPAFULL als mit den Dokumenten der Datenbank PATDPA gearbeitet werden soll. Außerdem könnte der Einfluss einer anfragespezifischen Stoppwortliste auf das Clustering-Ergebnis überprüft werden, um zu ermitteln, ob deren Anwendung mit oder ohne Termgewichtung zu besseren Ergebnissen führt. Für ein Software-Tool, das Endnutzern zum Clustern von Patentdokumenten zur Verfügung gestellt werden könnte, ist eine Festlegung der Clusteranzahl durch den Nutzer wünschenswert, so dass dieser explorativ die für ihn und sein Informationsbedürfnis geeignete Clusteranzahl frei wählen kann. Außerdem wäre die Erzeugung von aussagekräftigen Benennungen der Cluster als Hilfe für den Nutzer denkbar, wozu wiederum verschiedene Ansätze und Verfahren auf ihre Effektivität hin untersucht werden müssten.

Sollen Ergebnismengen auf eine Datenbank-Suchanfrage automatisch gruppiert werden, wie es z.B. bei der Patentdaten-Recherche denkbar ist, müssen dazu Clustering-Verfahren eingesetzt werden. In dieser Arbeit wurden verschiedene Clustering-Verfahren verglichen, um als Ergebnis eine Tendenzaussage zur Eignung eines bestimmten Verfahrens zu erhalten. Weitergehende Untersuchungen mit Nutzerbefragungen müssten folgen, um zu einer abschließenden Beurteilung zu gelangen, ob durch den Einsatz von Clustering-Verfahren eine wie im einleitenden Kapitel beschriebene Komplexitätsreduktion für den Anwender tatsächlich zu ermöglichen ist.

Literaturverzeichnis

[Anderberg 1972] ANDERBERG, Michael R.: *Cluster Analysis for Applications.* New York, San Francisco, London : Academic Press, 1972

[Backhaus et al. 2003] BACKHAUS, Klaus ; ERICHSON, Bernd ; PLINKE, Wulff ; WEIBER, Rolf: *Multivariate Analysemethoden. Eine anwendungsorientierte Einführung.* 10. Auflage. Berlin et al. : Springer-Verlag, 2003

[Bauer und Schneider 1990] BAUER, Gabi ; SCHNEIDER, Christine: *Analyse der Texterschließung.* S. 34 – 51. In: KRAUSE, Jürgen (Hrsg.) ; WOMSER-HACKER, Christa (Hrsg.): *Das Deutsche Patentinformationssystem. Entwicklungstendenzen, Retrievaltests und Bewertungen.* Köln, Berlin, Bonn, München : Carl Heymanns Verlag, 1990

[Belew 2000] BELEW, Richard K.: *Finding Out About. A Coginitive Perspective on Search Engine Technology and the WWW.* Cambridge : Cambridge University Press, 2000

[Bergmann 2004] BERGMANN, Ralph: *Unterlagen zur Vorlesung „Wissensentdeckung und Maschinelles Lernen", § 7 Clusteranalyse.* 2004. – Universität Hildesheim, Gruppe Daten- und Wissensmanagement

[Berkhin 2002] BERKHIN, Pavel: Survey Of Clustering Data Mining Techniques / Accrue Software. San Jose, CA, 2002. – Forschungsbericht. – URL `http://citeseer.nj.nec.com/berkhin02survey.html`. – Zugriffsdatum: 14.08.2004, 16:15 Uhr MEZ

[Bortz 1989] BORTZ, Jürgen: *Statistik für Sozialwissenschaftler.* Berlin et al. : Springer-Verlag, 1989

[Cooper 1988] COOPER, William S.: Getting beyond Boole. In: *Inf. Process. Manage.* 24 (1988), Nr. 3, S. 243–248. – ISSN 0306-4573

[Cutting et al. 1992] CUTTING, Douglass R. ; PEDERSEN, Jan O. ; KARGER, David ; TUKEY, John W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In: *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, URL `http://citeseer.ist.psu.edu/cutting92scattergather.html`. – Zugriffsdatum: 05.10.2004, 19:48 Uhr MEZ, 1992, S. 318–329

[Day 1996] DAY, H. E.: *Complexity theory: An introduction for practitioners of classification.* S. 190 – 211. In: ARABIE, P. (Hrsg.) ; HUBERT, J. (Hrsg.) ; DE SOETE, G. (Hrsg.): *Clustering and Classification.* Singapore, New Jersey, London : World Scientific Publishing, 1996

[Deichsel und Trampisch 1980] DEICHSEL, G. ; TRAMPISCH, H. J.: *Clusteranalyse und Diskriminanzanalyse.* Stuttgart : Gustav Fischer Verlag, 1980

[El-Hamdouchi und Willet 1989] EL-HAMDOUCHI, A. ; WILLET, P.: Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval. In: *The Computer Journal* 32 (1989), Nr. 3, S. 220–227

[Ertöz et al. 2002] ERTÖZ, Levent ; STEINBACH, Michael ; KUMAR, Vipin: *A New Shared Nearest Neighbor Clustering Algorithm and its Applications.* 2002. – URL `http://www-users.cs.umn.edu/~kumar/papers/siam_hd_snn_cluster.pdf`. – Zugriffsdatum: 14.08.2004, 16.08 Uhr MEZ

[Ertöz et al. 2003a] ERTÖZ, Levent ; STEINBACH, Michael ; KUMAR, Vipin: *Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach.* S. 83–103. In: WU, Weili (Hrsg.) ; XIONG, Hui (Hrsg.) ; SHEKHAR, Shashi (Hrsg.): *Clustering and Information Retrieval.* Dordrecht : Kluwer Academic Publishers, 2003

[Ertöz et al. 2003b] ERTÖZ, Levent ; STEINBACH, Michael ; KUMAR, Vipin: *Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach.* 2003. – URL `http://www-users.cs.umn.edu/~kumar/papers/snn14.pdf`. – Zugriffsdatum: 06.10.2004, 11:11 Uhr MEZ

[Everitt et al. 2001] EVERITT, Brian S. ; LANDAU, Sabine ; LEESE, Morven: *Cluster Analysis.* Fourth Edition. London : Arnold, 2001

[Fattori et al. 2003] FATTORI, Michele ; PEDRAZZI, Giorgio ; TURRA, Roberta: Text mining applied to patent mapping: a practical business case. In: *World Patent Information* 25 (2003), Nr. 4, S. 335–342

[Ferber 2003] FERBER, Reginald: *Information Retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web.* Heidelberg : dpunkt.verlag, 2003

[FIZ-Karlsruhe 2000] FIZ-KARLSRUHE: *Im Zentrum des Wissens. Information als Dienstleistung.* 2000. – URL `http://www.fiz-karlsruhe.de/about_fiz/image-dt.pdf`. – Zugriffsdatum: 06.10.2004, 10:47 Uhr MEZ. – FIZ-Karlsruhe, Gesellschaft für wissenschaftliche Information mbH

[Göbel] GÖBEL, Heike: *Kurs zu „Patentrecherchen im Internet“.* – URL `http://www.uni-jena.de/chemie/ivs/patente/`. – Zugriffsdatum: 06.05.2004, 10:45 Uhr MEZ. – Informationsvermittlungsstelle (IVS) der Chemisch-Geowissenschaftlichen Fakultät der Friedrich-Schiller-Universität Jena. Ohne Jahresangabe.

[Gerstl et al. 2001] GERSTL, Peter ; HERTWECK, Matthias ; KUHN, Birgit: Text Mining: Grundlagen, Verfahren und Anwendungen. In: *HDM: Praxis der Wirtschaftsinformatik* (2001), Dezember, Nr. 222, S. 38–48

[Haenelt 2003] HAENELT, Karin: *Clustering. Kursfolien*. 2003. – URL `http://kontext.fraunhofer.de/haenelt/kurs/folien/Clustering.pdf`. – Zugriffsdatum: 14.08.2004, 16.05 Uhr MEZ

[Han und Kamber 2001] HAN, Jiawei ; KAMBER, Micheline: *Data Mining – Concepts and Techniques*. London : Academic Press, 2001

[Hearst und Pedersen 1996] HEARST, Marti A. ; PEDERSEN, Jan O.: Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. (1996), S. 76–84. – URL `http://citeseer.ist.psu.edu/hearst96reexamining.html`. – Zugriffsdatum: 05.10.2004, 19:40 Uhr MEZ

[Hösel und Walcher] HÖSEL, Volker ; WALCHER, Sebastian: *Clustering Techniques: A Brief Survey*. – URL `http://citeseer.ist.psu.edu/444077.html`. – Zugriffsdatum: 14.08.2004, 16:14 Uhr MEZ

[Jain und Dubes 1988] JAIN, A. K. ; DUBES, R. C.: *Algorithms for Clustering Data*. Upper Saddle River, NJ : Prentice-Hall, 1988

[Jain et al. 1999] JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data clustering: a review. In: *ACM Computing Surveys* 31 (1999), Nr. 3, S. 264–323. – URL `http://citeseer.ist.psu.edu/jain99data.html`. – Zugriffsdatum: 14.08.2004, 16:12 Uhr MEZ

[Kamps et al. 2004] KAMPS, Jaap ; MONZ, Christof ; RIJKE, Maarten de ; SIGURBJÖRNSSON, Börkur: Approaches to Robust and Web Retrieval. (2004), S. 594–600. – URL `http://www.science.uva.nl/~mdr/Publications/Files/trec-2003-rbwb-proceedings.pdf`. – Zugriffsdatum: 06.05.2004, 10:56 Uhr MEZ. – Language & Inference Technology Group, University of Amsterdam

[Karypis 2003] KARYPIS, George: CLUTO. A Clustering Toolkit. (Release 2.1.1) / University of Minnesota. Department of Computer Science. URL `http://www.cs.umn.edu/~karypis/cluto`. – Zugriffsdatum: 15.08.2004, 10:30 Uhr MEZ, 2003 (#02-017). – Forschungsbericht

[Kaufmann und Pape 1984] KAUFMANN, Heinz ; PAPE, Heinz: *Clusteranalyse*. S. 371–472. In: FAHRMEIR, Ludwig (Hrsg.) ; HAMMERLE, Alfred (Hrsg.): *Multivariate statistische Verfahren*. Berlin : de Gruyter, 1984

[Krause 1987] KRAUSE, Jürgen: *Problemfeld Patenterteilung und derzeitge Informationsbeschaffung*. S. 208–233. In: KRAUSE, Jürgen (Hrsg.): *Inhaltserschließung von Massendaten. Zur Wirksamkeit informationslinguistscher Verfahren am Beispiel des Deutschen Patentinformationssystems*. Hildesheim, Zürich, New York : Olms, 1987

[Kumar 2003] KUMAR, Vipin: *Cluster Analysis: Basic Concepts and Algorithms*. 2003. – URL http://www-users.cs.umn.edu/~kumar/csci5980/lecture/ch7.pdf. – Zugriffsdatum: 20.10.2004, 10.00 Uhr MEZ. – Textbook for course Data Mining (Spring 2004) at the University of Minnesota

[Kural et al. 1999] KURAL, Yasemin ; ROBERTSON, Steve ; JONES, Susan: *Clustering Information Retrieval Search Outputs*. 1999. – URL http://ewic.bcs.org/conferences/1999/21stirsg/papers/paper9.pdf. – Zugriffsdatum: 05.10.2004, 19:50 Uhr MEZ. – 21st Annual BCS-IRSG Colloquium on IR

[Kural et al. 2001] KURAL, Yasemin ; ROBERTSON, Steve ; JONES, Susan: Deciphering cluster representations. In: *Information Processing and Management* 37 (2001), Nr. 4

[Larsen und Aone 1999] LARSEN, Bjornar ; AONE, Chinatsu: Fast and effective text mining using linear-time document clustering. In: *Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining* (1999), S. 16–22

[Ludwig 1994] LUDWIG, Michaela: *Statistische Verfahren zur Ermittlung von Ähnlichkeitsbeziehungen am Beispiel von Werkstoffdaten*, Universität Regensburg. Philosophische Fakultät IV (Sprach- und Literaturwissenschaften), Diplomarbeit, 1994

[Maarek et al. 2002] MAAREK, Yoëlle S. ; FAGIN, Ronald ; BEN-SHAUL, Israel Z. ; PELLEG, Dan: *Ephemeral Document Clustering for Web Applications*. August 2002. – URL http://citeseer.ist.psu.edu/maarek00ephemeral.html. – Zugriffsdatum: 05.10.2004, 19:38 Uhr MEZ. – IBM Research Report RJ 10186

[Macskassy et al. 1998] MACSKASSY, Sofus A. ; BANERJEE, Arunava ; DAVISON, Brian D. ; HIRSH, Haym: Human Performance on Clustering Web Pages / Department of Computer Science Rutgers, The State University of New Jersey. URL ftp://www.cs.rutgers.edu/pub/technical-reports/dcs-tr-355.ps.Z. – Zugriffsdatum: 06.10.2004, 11:23 MEZ, 1998 (DCS-TR-355). – Forschungsbericht

[Mandl und Koelle 2001] MANDL, Thomas ; KOELLE, Ralph: *Kapitel Clustering. Vorlesung Data Mining.* 2001. – URL http://www.uni-hildesheim.de/~mandl/Lehre/DataMining_SS01/DataMining_04_Clustering.pdf. – Zugriffsdatum: 26.10.2004, 15.35 Uhr MEZ. – Universität Hildesheim

[Manning und Schütze 2002] MANNING, Christopher D. ; SCHÜTZE, Hinrich: *Foundations of statistical natural language processsing*. Cambridge, Massachusetts, London : MIT Press, 2002. – Second Printing with corrections, 2000

[Milligan 1996] MILLIGAN, Glenn W.: *Clustering Validation: Results and implications for applied analyses*. S. 341–375. In: ARABIE, P. (Hrsg.) ; HUBERT, J. (Hrsg.) ; DE SOETE, G. (Hrsg.): *Clustering and Classification*. River Edge, NJ : World Scientific Publishers, 1996

[Neto et al. 2000] NETO, J. ; SANTOS, A. ; KAESTNER, C. ; FREITAS, A.: *Document clustering and text summarization*. 2000. – URL http://citeseer.ist.psu.edu/laroccaneto00document.html. – Zugriffsdatum: 09.10.2004, 13:00 Uhr MEZ

[Panyr 1986] PANYR, Jiri: *Automatische Klassifikation und Information Retrieval: Anwendung und Entwicklung komplexer Verfahren in Information-Retrieval-Systemen und ihre Evaluierung*. Tübingen : Niemeyer, 1986

[Patentgesetz] *Patentgesetz*. – URL http://bundesrecht.juris.de/bundesrecht/patg/__9.html. – Zugriffsdatum: 15.11.2004, 10:41 Uhr MEZ. – Verkündungsfundstelle: RGBl II 1936, 117, Stand: Neugefasst durch Bek. v. 16.12.1980; 1981 I 1, zuletzt geändert durch Art. 4 Abs. 41 G v. 5.5.2004 I 718

[Pinker 1997] PINKER, Stephen: *How the Mind Works*. New York : Norton, 1997

[Popescul und Ungar 2000] POPESCUL, A. ; UNGAR, L.: *Automatic Labeling of Document Clusters*. 2000. – URL http://citeseer.ist.psu.edu/popescul00automatic.html. – Zugriffsdatum: 05.10.2004, 19:47 Uhr MEZ

[Rasmussen 1992] RASMUSSEN, Edie: *Clustering Algorithms*. In: FRAKES, William B. (Hrsg.) ; RICARDO, Baeza-Yates (Hrsg.): *Data Structures and Algorithms*, Prentice-Hall, 1992

[Rasmussen und Karypis 2004] RASMUSSEN, Matt ; KARYPIS, George: gCLUTO – An Interactive Clustering, Visualization, and Analysis System / University of Minnesota. Department of Computer Science and Engineering. URL http://www-users.cs.umn.edu/~karypis/publications/Papers/PDF/gCLUTO.pdf. – Zugriffsdatum: 06.10.2004, 10:42 MEZ, 2004 (#04–021). – Forschungsbericht

[van Rijsbergen 1979] RIJSBERGEN, C. J. van: *Information Retrieval*. Second Edition. London : Butterworths, 1979

[Robertson et al. 2000] ROBERTSON, S. E. ; S., Walker ; BEAULIEU, M.: Experimentation as a way of life: Okapi at TREC. In: *Information Processing and Management* 36 (2000), Nr. 1, S. 95–108

[Robertson und Walker 2000] ROBERTSON, S. E. ; WALKER, S.: Okapi/Keenbow at TREC-8, NIST Special Publication 500-264, 2000, S. 151–161

[Schramm 2004] SCHRAMM, Reinhard: *PATON-Vorlesungsreihe*. 2004. – URL http://www.paton.tu-ilmenau.de/lehre/vorlesung/. – Zugriffsdatum: 14.10.2004, 22.00 Uhr MEZ

[Statsoft] STATSOFT: *Cluster Analysis*. – URL http://www.statsoft.com/textbook/stcluan.html. – Zugriffsdatum: 10.10.2004, 22:55 Uhr MEZ

[Stein et al. 2003] STEIN, Benno ; EISSEN, Sven Meyer zu ; WISSBROCK, Frank: On Cluster Validity and the Information Need of Users. Benalmádena, Spain : ACTA Press, September 2003, S. 216–221

[Steinbach et al. 2002] STEINBACH, Michael ; ERTÖZ, Levent ; KUMAR, Vipin: *The Challenges of Clustering High Dimensional Data*. 2002. – URL http://www-users.cs.umn.edu/~ertoz/papers/clustering_chapter.pdf. – Zugriffsdatum: 14.08.2004, 16.08 Uhr MEZ

[Steinbach et al. 2000] STEINBACH, Michael ; KARYPIS, George ; KUMAR, Vipin: A comparison of document clustering techniques / University of Minnesota. Department of Computer Science and Engineering. URL http://citeseer.ist.psu.edu/steinbach00comparison.html. – Zugriffsdatum: 14.08.2004, 16:11 Uhr MEZ, 2000 (#00–034). – Forschungsbericht

[Steinhausen und Langer 1977] STEINHAUSEN, Detlef ; LANGER, Klaus: *Clusteranalyse: Einführung in Methoden und Verfahren der automatischen Klassifikation*. Berlin, New York : de Gruyter, 1977

[Thomä und Tribiahn 2002] THOMÄ, Elke ; TRIBIAHN, Rudolf: *Leitfaden für Patentrecherchen mit STN EASY*. 2002. – URL http://www.stn-international.de/training_center/patents/patguide/easy_de/EasyGuide.pdf. – Zugriffsdatum: 14.08.2004, 16.05 Uhr MEZ. – Informationsvermittlungsstelle (IVS) der Chemisch-Geowissenschaftlichen Fakultät der Friedrich-Schiller-Universität Jena

[Trippe 2003] TRIPPE, Anthony J.: Patinformatics: Tasks to tools. In: *World Patent Information* 25 (2003), Nr. 4, S. 211–221

[TU Ilmenau] TU Ilmenau (Veranst.): *Leitfaden zu STN-Patentdatenbanken*. – URL http://www.patent-inf.tu-ilmenau.de/schulungszentrum/guide_de_02/gd02_de_pdf/Textrecherchen.pdf. – Zugriffsdatum: 10.08.2004, 21:30 Uhr MEZ

[Vogel 1975] VOGEL, Friedrich: *Probleme und Verfahren der numerischen Klassifikation*. Göttingen : Vandenhoeck und Ruprecht, 1975

[Wahrig 2000] WAHRIG, Gehrhard ; WAHRIG-BURFEIND, Renate (Hrsg.): *Deutsches Wörterbuch*. Gütersloh, München : Bertelsmann-Lexikon Verlag, 2000

[Walz 2001] WALZ, Guido (Hrsg.): *Lexikon der Mathematik – Band 3*. Heidelberg : Spektrum Akad. Verlag, 2001

[Witten und Frank 2000] WITTEN, Ian H. ; FRANK, Eibe: *Data Mining: Practical machine learning tools with Java implementations*. San Francisco : Morgan Kaufmann, 2000

[Wittmann 1992] WITTMANN, Alfred: *Grundlagen der Patentinformation und Patentdokumentation*. Berlin, Offenbach : vde-verlag, 1992

[Womser-Hacker 2003] WOMSER-HACKER, Christa: *Kapitel Modelle. Vorlesung Information Retrieval in Theorie und Praxis*. 2003. – URL http://www.

uni-hildesheim.de/media/ifas/IR_Vorlesung_3.pdf. – Universität Hildesheim

[Wurzer 2003] WURZER, Alexander J.: *Wettbewerbsvorteile durch Patentinformationen*. 2. überarbeitete Auflage. Karlsruhe : FIZ-Karlsruhe, 2003

[Zamir und Etzioni 1998] ZAMIR, Oren ; ETZIONI, Oren: Web Document Clustering: A Feasibility Demonstration. In: *Research and Development in Information Retrieval*, URL http://citeseer.ist.psu.edu/zamir98web.html, 1998, S. 46–54

[Zamir et al. 1997] ZAMIR, Oren ; ETZIONI, Oren ; MADANI, Omid ; KARP, Richard M.: Fast and Intuitive Clustering of Web Documents. In: *Knowledge Discovery and Data Mining*, URL http://citeseer.ist.psu.edu/article/zamir97fast.html. – Zugriffsdatum: 05.10.2004, 19:48 Uhr MEZ, 1997, S. 287–290

[Zhao und Karypis 2001] ZHAO, Ying ; KARYPIS, George: *Criterion functions for document clustering: Experiments and analysis*. 2001. – URL http://citeseer.ist.psu.edu/zhao02criterion.html. – Zugriffsdatum: 14.08.2004, 16.09 Uhr MEZ

[Zhao und Karypis 2002] ZHAO, Ying ; KARYPIS, George: Evaluation of hierarchical clustering algorithms for document datasets / University of Minnesota. Department of Computer Science and Engineering. URL citeseer.ist.psu.edu/zhao02evaluation.html. – Zugriffsdatum: 08.10.2004, 13:00 Uhr MEZ, 2002 (#02-022.). – Forschungsbericht

[Zhao und Karypis 2003] ZHAO, Ying ; KARYPIS, George: Hierarchical Clustering Algorithms for Document Datasets / University of Minnesota. Department of Computer Science and Engineering. URL https://wwws.cs.umn.edu/tech_reports_upload/tr2003/03-027.pdf. – Zugriffsdatum: 08.10.2004, 13:01 Uhr MEZ, 2003 (#03-027.). – Forschungsbericht

[Zhao und Karypis 2004] ZHAO, Ying ; KARYPIS, George: *Soft Clustering Criterion Functions for Partitional Document Clustering*. 2004. – URL http://citeseer.ist.psu.edu/zhao02criterion.html. – Zugriffsdatum: 14.08.2004, 16.09 Uhr MEZ

Anhang A Eingesetzte Software zur Durchführung der Clustering-Experimente

A.1 CLUTO

A.1.1 Herkunfts- und Lizenzinformationen

CLUTO (**Clu**stering **To**olkit) ist eine Software, die an der University of Minnesota im Department of Computer Science von George Karypis entwickelt wurde. Im dazugehörigen Handbuch wird die Software wie folgt charakterisiert: „CLUTO is a software package for clustering low and high dimensional datasets and for analyzing the characteristics of the various clusters." (Karypis 2003, 4) Für die Experimente wurde die aktuelle Version 2.1.1 mit Stand vom 28.11.2003 gewählt. Die Software ist für den Einsatz in Forschung und Lehre durch Non-Profit Organisationen lizenziert. Andere Einrichtungen dürfen die Software zu Evaluationszwecken testen, ein darüber hinausgehender Einsatz erfordert die Zustimmung der Lizenzinhaber (Regents of the University of Minnesota). Die Software liegt als Binärdatei vor, die unter `http://www.cs.umn.edu/~karypis/cluto` zu beziehen ist. Eine Veröffentlichung des Quellcodes (in ANSI C) ist für die nachfolgenden Versionen beabsichtigt (vgl. Karypis 2003, 71). Außer den ausführbaren Dateien liegt der CLUTO-Distribution eine Bibliothek (libcluto.lib) und eine Header-Datei (cluto.h) bei, um die Funktionalität von CLUTO in eigene C oder C++ Programme einzubinden (vgl. Karypis 2003, 37). Eine sehr ausführliche Dokumentation in Form eines Handbuches liegt der Software bei (Karypis 2003).

A.1.2 Möglichkeiten der Software

Im Programmpaket sind verschiedene Arten von Clustering-Algorithmen realisiert (hierarchische, partitionierende, graph-basierte Algorithmen), die mit zahlreichen verschiedenen Gütefunktionen (z.B. Single Linkage oder UPGMA bei hierarchischen Verfahren) die Ausgangsdaten clustern. Die erzeugten Lösungen können graphisch in Form von Dendrogrammen oder einer Art Instanz-zu-Cluster-Darstellung (Matrixdarstellung) visualisiert werden. Zur Analyse der Ergebnisse können interne und externe Bewertungsmaße angezeigt werden (vgl. Kapitel 9). Sämtliche Parameter, wie Anzahl der zu bestimmenden Cluster, Ausgangsdaten, Art des Clustering-Verfahrens

oder Speicherort für Cluster-Lösungen, werden beim Programmstart über Kommandozeilenargumente übergeben. Um den Nutzern ein verbessertes Interface für die Bedienung bereit zu stellen, entwickelten Matt Rasmussen, Mark Newman und George Karypis einen graphischen Aufsatz, den sie „gCLUTO“ nannten. Er ist in der Version 1.0 (Stand vom 19.11.2003) unter der URL `http://www.cs.umn.edu/~mrasmus/gcluto` zu beziehen, eine Beschreibung des Programms ist in dem Artikel von Rasmussen und Karypis (2004) zu finden.

A.1.3 Format der Eingabedaten

Matrix-Datei (*.mat) Die zu clusternden Daten müssen in Form einer Matrix-Datei vorliegen, die den folgenden Aufbau hat (vgl. Karypis 2003, 29 ff.): Jede Zeile der Datei entspricht einer Instanz, wobei die Spalten die Dimensionen oder Merkmale der Instanz beschreiben. Für die Experimente wurde das „sparse“-Format (engl. sparse = dünn, verstreut) verwendet, bei dem nur die Dimensionen aufgeführt werden, die einen Wert größer 0 aufweisen (das Gegenteil wäre das „dense“-Format, bei dem sämtliche Dimensionen, auch mit Wert gleich 0, aufgezählt würden). In der allerersten Zeile befinden sich zwingend in dieser Reihenfolge Angaben zur Matrix-Datei: Erstens die Gesamtzahl der Instanzen (n), zweitens die maximale Anzahl der Dimensionen (m) und drittens die Gesamtzahl der Einträge in der $n \times m$ Matrix , die ungleich 0 sind. Die einzelnen Merkmale einer Instanz werden durch Wertepaare beschrieben, die durch Leerzeichen voneinander getrennt sind. Der erste Wert gibt die Spalte bzw. das Merkmal an, das die darauf folgend genannte Ausprägung annimmt (vgl. Abbildung A.1). Im vorliegenden Anwendungsgebiet stellen die Merkmale die Terme einer Kollektion von Patentdokumenten dar, die Ausprägungen stellen die Häufigkeiten der Terme innerhalb einer Instanz dar.

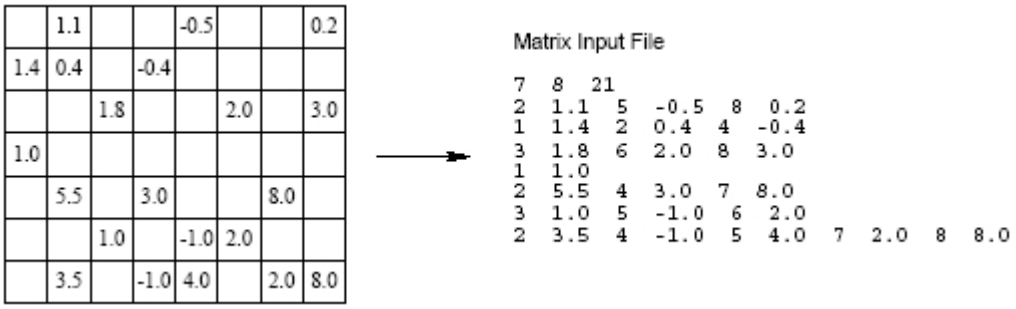


Abbildung A.1: Format der Eingabedaten (Karypis 2003, 33)

Bezeichnung der Merkmale (*.clabel) Die Datei „column label“ (clabel) beinhaltet eine Bezeichnung für jedes Merkmal (hier: Term). Besitzt die Datenmatrix m Merkmale, so besteht die zugehörige *.clabel-Datei aus m Zeilen, wobei der i-ten Zeile das i-te Merkmal entspricht (vgl. Karypis 2003, 33).

Bezeichnung für Instanzen (*.rlabel) Jeder Instanz kann eine Bezeichnung (z.B. die Dokumenten-ID) zugewiesen werden („row label file“). Beinhaltet die Datenmatrix n Instanzen, so besteht die zugehörige *.rlabel-Datei aus n Zeilen, wobei die n-ten Zeile die n-te Instanz beschreibt (vgl. Karypis 2003, 33).

Klassenzuordnung (*.rclass) Jeder Instanz kann, wenn bekannt oder vorhanden, eine zuvor festgelegte Klasse zugeordnet werden. Dies geschieht mit Hilfe der Datei *.rclass („row class“), die die Instanzen einer zugehörigen Matrix-Datei beschreibt. Besitzt die Matrix-Datei n Instanzen, so beinhaltet die Datei *.rclass n Zeilen, wobei die n-te Zeile die n-te Instanz charakterisiert (vgl. Karypis 2003, 34).

A.2 WEKA

A.2.1 Herkunfts- und Lizenzinformationen

Das Programmpaket WEKA (**W**aikato **E**nvironment for **K**nowledge **A**nalysis) ist eine Software, die unter der GNU General Public License frei einsetzbar ist und deren Quellcode veröffentlicht ist. In dieser Programm-Suite sind die verschiedensten Algorithmen rund um den Themenbereich „Maschinelles Lernen“ ausschließlich in JAVA implementiert. Entwickelt wurde die Software an der University of Waikato in Neuseeland und gelangte durch die Buchpublikation von Witten und Frank (2000) zu großer Bekanntheit. Systemvoraussetzung ist das Vorhandensein einer JAVA Virtual Machine ab Version 1.4. Für die Experimente im Zuge der Magisterarbeit wurde die Version 3.4.1 von Weka eingesetzt. Eine Dokumentation liegt hauptsächlich in Form einer Javadoc vor, welche die API des Quellcodes beschreibt. Zu Beziehen ist die Software unter der URL `http://www.cs.waikato.ac.nz/~ml/weka/`.

A.2.2 Möglichkeiten der Software

Die Software besitzt mehrere GUIs über die die Algorithmen und deren Parameter gesteuert werden können. Zudem besteht die Möglichkeit, die Algorithmen direkt durch Aufruf der bereitgestellten JAVA-Methoden im eigenen Quellcode zu nutzen. Weka beinhaltet Werkzeuge zur Vorverarbeitung von Daten, Klassifikation, Regressions-Analyse, Clustering, Assoziationsregeln und Möglichkeiten zur Visualisierung der Ergebnisse.

A.2.3 Format der Eingabedaten

Die Eingabedaten müssen im so genannten ARFF-Format vorliegen (Attribute Relation File Format), das im Rahmen des Weka-Projekts als Datenformat entwickelt wurde[1]. In einer ARFF-Datei werden im Abschnitt „@Relation“ zuerst die Attribute (hier: Terme) und deren Datentyp (hier: numerisch, da gewichtete Termfrequenzen vorliegen) deklariert. Im zweiten Abschnitt der ARFF-Datei, dem „@Data“-Abschnitt, werden die Ausprägungen der Merkmale durch Kommas getrennt erfasst. Die Reihenfolge der Merkmale entspricht genau derjenigen im Abschnitt „@Relation“.

A.3 SNN-Algorithmus

A.3.1 Herkunfts- und Lizenzinformationen

Der Algorithmus SNN wurde erstmals im Artikel von Ertöz et al. (2002) vorgestellt. Er liegt im Quellcode in Form eines in C++ geschriebenen Programms vor, das mit Stand vom 03.04.2002 von der Homepage des Entwicklers (Levent Ertöz) heruntergeladen werden kann („Finally, we have made our SNN clustering algorithm publicly available so that others can try it for themselves. It can be download from `http://www.cs.umn.edu/~ertoz/snn/`“ (Ertöz et al. 2003b, 12)). Es werden keinerlei Lizenzinformationen angegeben. Eine Dokumentation der Programmoptionen findet sich in der der Software beigefügten Readme-Datei.

A.3.2 Möglichkeiten der Software

Das Programm wird zusammen mit den gewählten Parametern auf der Kommandozeile gestartet. Es besitzt keinerlei Zusatzfunktionalität (wie z.B. die Generierung von Statistiken). Um den Inhalt der Ausgabedatei analog zu den CLUTO-Ausgabedateien zu formatieren, wurde der Quelltext für diese Arbeit abgeändert. Im (meist) größten Cluster mit der Nummer 0 in der Ergebnisdatei wurden die Instanzen gesammelt, die nicht einem Cluster zugeordnet werden konnten.

A.3.3 Format der Eingabedaten

Die Formate der Eingabedaten entsprechen im Aufbau denen von CLUTO: eine Matrix-Datei (*.mat) und eine Datei, die die Bezeichnungen für die Instanzen (*.rname) enthält (entspricht der *.rlabel-Datei bei CLUTO). Außerdem kann eine vorgefertigte

[1] Eine detailliertere Beschreibung des Datenformats findet sich unter `http://www.cs.waikato.ac.nz/~ml/weka/arff.html`

Klassenzuweisung, wenn vorhanden, angegeben werden (hier: *.rlabel; entspricht bei CLUTO der *.rclass-Datei).

A.4 Autoclass-C

A.4.1 Herkunfts- und Lizenzinformationen

Autoclass-C ist eine Public-Domain Version der Software Autoclass III. Sie wurde von Dr. Diane Cook und Joseph Potts von der Universität Arlington in Texas programmiert und durch Will Taylor getestet, dokumentiert und als Paket zusammengestellt, das unter `http://ic.arc.nasa.gov/projects/bayes-group/autoclass/autoclass-c-program.html` heruntergeladen werden kann. Die Experimente wurden mit der aktuellsten Version (V3.3.4 vom 24.01.2002) durchgeführt.

A.4.2 Möglichkeiten der Software

Das Programm verfolgt einen probabilistischen Clustering-Ansatz (siehe Kapitel 7.3). Sowohl numerische als auch nominale Attribute können verarbeitet werden, wobei für die numerischen Attribute eine Gauss-Verteilung als Modell der Mischverteilung angenommen wird. Autoclass-C ermittelt die optimale Anzahl der Cluster automatisch, indem der EM-Algorithmus mit wechselnden Initialwerten wiederholt ausgeführt wird und die „beste“ Lösung weiterverwendet wird. Das Programm ist sehr rechenintensiv; mit steigender Anzahl der Attribute und Instanzen steigt die benötigte Rechenzeit sehr stark an.

A.4.3 Format der Eingabedaten

Die Attribute werden in der ***.hd2-Datei** zunächst deklariert und mit einem Typ versehen. Für die Experimente wurde der Typ „scalar“ mit einem Nullpunkt bei 0 und in Anlehnung an die Experimente von Neto et al. (2000, 6) einem Messfehler pro Attribut in der Größenordnung von 0,1% zugewiesen (rel_error = 0.001). „The fundamental question in all of this is: ‚To what extent do you believe the numbers that are to be given to Autoclass?‘ “ (Dokumentation zum Autoclass-C Paket, Datei: preparations-c.txt). In der ***.db2-Datei** wird die Datenmatrix gespeichert, wobei eine Zeile einem Objekt entspricht und dessen Attribute durch Kommata getrennt aufgezählt werden.

Anhang B Im Rahmen der Magisterarbeit entwickelte Software

Alle Anwendungen und Tools, die im Rahmen dieser Magisterarbeit entwickelt wurden, wurden unter JAVA 1.4.2 programmiert und getestet. Die Programmiersprache JAVA wurde aus Gründen der Plattformunabhängigkeit gewählt. Auf Folgendes wird hingewiesen:

- Zum Nutzen der Programme muss ein JAVA Runtime-Environment ab Version 1.4. vorhanden sein, da Reguläre Ausdrücke verwendet werden, die erst ab dieser Version offizieller Bestandteil des Sprachumfangs von JAVA sind.
- Sämtliche Software, die für diese Magisterarbeit vom Autor entwickelt wurde, hat einen experimentellen Charakter (Alpha Version). Mögliche Fehler – soweit bekannt und soweit möglich – werden abgefangen; eine 100%ige Fehlerfreiheit wird nicht garantiert.

B.1 Pre-Processing-Tool PatentPreProcess

B.1.1 Programmeigenschaften und -fähigkeiten

Die bei einer Patentrecherche über STN zurückgelieferten Patentdokumente müssen zur Nutzung durch die Clustering-Verfahren erst aufbereitet werden und in ein für die Programme geeignetes Eingabeformat konvertiert werden. Als Eingabedaten für das Tool `PatentPreProcess.java` werden Textdateien vorausgesetzt, die die Antwortdokumente auf jeweils eine Suchanfrage an die Datenbank PATDPA beinhalten (im Format „brief“).

Zum Auslesen der Informationen, die in den Datenbankfeldern hinterlegt sind, wird auf die JAVA-Klasse `PatentParser.java` zurückgegriffen, die hierfür leicht modifiziert werden musste. Sie entstand als Teilprojekt der studentischen Gruppe (1a) anlässlich des Projekt-Seminars „Semantic Web und Ontologien“ (Wintersemester 2003/2004, Universität Hildesheim) unter der Leitung von Diplom-Informationswissenschaftler Robert Strötgen, Dipl.-Inform. Ralph Koelle und Dr. René Schneider.

Das Programm liest die Patentdokumente ein, berechnet Term- und Kollektionsfrequenz, führt eine Gewichtung der Terme durch und erstellt die Eingabedateien für

die verschiedenen Clustering-Tools. Zusätzlich wird eine Statistik über die konvertierten Dokumente erstellt, die als CSV-Datei (Comma Separated Value-Datei) zur Weiterverarbeitung z.B. durch Import in Microsoft Excel bereitsteht.

Folgende Ausgabedateien werden in den weiter unten beschriebenen Verzeichnissen erstellt, wobei diese Verzeichnisse als Unterverzeichnisse des Basisverzeichnisses angelegt werden, das wiederum bei der Konfiguration mittels des Parameters `targetDirectory` festgelegt wurde:

- `\arff:` Eingabedateien für das Programmpaket WEKA im Format *.ARFF.
  Es werden sowohl Dateien im Format „dense“ und „sparse“ erzeugt. Die konvertierten Daten liegen in einer gewichteten (Dateiendung `_weighted.arff`), als auch in einer ungewichteten Variante vor.
- `\autoclass:` Eingabedateien für das Programm Autoclass-C.
  Diese Dateien liegen nur als gewichtete Daten vor.
- `\doc2mat:` Eingabeformat für das Perl-Script `doc2mat.pl`.
  Dieses wird von den Autoren von CLUTO zur Erzeugung der benötigten Eingabeformate vorgeschlagen. Es fand in der Anfangszeit der Vorab-Versuche Anwendung, jedoch ist damit beispielsweise keine Termgewichtung möglich, so dass diese Funktionalität selbst programmiert werden musste.
- `\mainIPC:` In diesem Verzeichnis wird eine „Pseudo-Clustering-Lösung“ erstellt, d.h. die Patentdokumente einer Anfrage werden anhand ihrer MainIPC in Gruppen eingeteilt. Das Ergebnisformat entspricht dem von CLUTO bzw. SNN.
- `\mat:` Eingabedateien für das Programm CLUTO.
  Die konvertierten Daten liegen sowohl in einer gewichteten (Dateiendung `_weighted.mat`), als auch in einer ungewichteten Variante vor.
- `\vectors:` Hier befinden sich die aus den Textdateien mit Patentdokumenten extrahierten Datenfelder, wobei diese Daten pro Anfrage in einer separaten Datei (`_PatentVector.dat`) gespeichert werden. Diese Dateien werden zur Anzeige der Patentdaten mittels *ExperimenterGUI* bzw. während der Evaluierung mittels *ClustEv* benötigt.

B.1.2 Konfiguration

Da sich das Tool zum Konvertieren und Vorverarbeiten der Patentdokumente noch im Entwicklungsstadium befindet, erfolgt die Konfiguration direkt in der *Main-Methode* des Quellcodes. Dort können folgende Parameter variiert werden:

- `sourceDirectory =` Pfad des Speicherorts der Eingabe-Textdateien
- `targetDirectory =` Pfad des Basisverzeichnis, in dessen Unterverzeichnisse die konvertierten Daten geschrieben werden
- `weightingScheme =` Termgewichtungsschema: ’okapi’ oder ’tfidf’

- k1 = <Int> Parameter für das Okapi-Gewichtungsschema
- b = <Int> Parameter für das Okapi-Gewichtungsschema
- minimumNumberOfTerms = <Int> Mindestanzahl an Termen pro Dokument nach Stemming und Stoppwort-Eliminiation
- queryTermsAsStopwords = *true*|*false*
  Gibt an, ob für eine Anfrage spezifische Stoppwörter zu einer allgemeinen Stoppwortliste hinzugefügt werden sollen. Diese anfragespezifischen Stoppwörter werden für jede Anfrage getrennt in einer Textdatei gespeichert (ohne Datei-Endung), die im gleichen Verzeichnis wie die JAVA-Klasse Patent-PreProcess.java liegen muss. Außerdem muss sie den gleichen Namen wie die Textdatei mit den Ausgangsdokumenten aufweisen und pro Zeile dieser Textdatei darf nur jeweils ein hinzuzufügendes Stoppwort vorkommen.
- addIPC = *true*|*false*
  Hinzufügen der IPC-Hauptklasse als Term für die Eingabedaten
- checkForDuplicates = *true*|*false*
  Anhand eines String-Vergleichs werden Patentfamilien-Doppel eliminiert, so dass nur ein Patent in die Datenbasis zum Clustern Eingang findet.

B.1.3 Statistiken

Es werden zwei Statistiken erstellt, die im Basisverzeichnis (targetDirectory) gespeichert werden: Die Verteilung der Patentdokumente einer Anfrage über die IPC-Klassen (distributionMainIPC.csv) sowie eine Gesamtstatistik aller verarbeiteten Anfragen (statistics.csv).

Statistik: Verteilung der Patentdokumente über die Klassen der IPC

Es werden folgende Informationen in dieser Statistik aufgeführt:

- Name der Anfrage (= Dateiname der Original-Textdatei im Quellverzeichnis sourceDirectory)
- Schlüssel der MainIPC-Klasse
- Anzahl der Patentdokumente, die zu dieser MainIPC-Klasse gehören
- Gesamtzahl der Patentdokumente dieser Anfrage

Statistik: Gesamtstatistik

Für alle verarbeiteten Anfragen wird eine Gesamtstatistik erstellt. Dazu werden für jede Anfrage folgende Informationen zusammengetragen.

- Name der Anfrage (= Dateiname der Original-Textdatei im Quellverzeichnis sourceDirectory)
- Gesamtzahl der Patentdokumente dieser Anfrage

- ❏ Anzahl der Dokumente, die nur das Feld TI (= Titel) aufweisen.
- ❏ Anzahl der Dokumente, die die Felder TI und AB (= Titel und Abstract) aufweisen.
- ❏ Anzahl der Dokumente, die die Felder TI, AB und MCLM (= Titel, Abstract und MainClaim) aufweisen.
- ❏ Anzahl der Dokumente, die die Felder TI und MCLM (= Titel und MainClaim) aufweisen.
- ❏ Anzahl der Terme
- ❏ höchste Anzahl an Termen, die ein Dokument dieser Anfrage aufweist.
- ❏ geringste Anzahl an Termen, die ein Dokument dieser Anfrage aufweist.
- ❏ durchschnittliche Anzahl an Termen, die ein Dokument dieser Anfrage enthält

B.1.4 Ablauf der Verarbeitung und Anmerkungen

Der Ablauf der gesamten Vorverarbeitung wird im Folgenden beschrieben (siehe Algorithmus 6):

Algorithmus 6: Ablauf der Verarbeitung (Pseudo-Code)

```
für alle Textdateien im Verzeichnis sourceDirectory, die Patentdokumente als
Ergebnis einer Anfrage beinhalten, tue
    für alle Patentdokumente, die zu einer Anfrage gehören, tue
        Lies ein Patentdokument ein.
        wenn Mindestanzahl an Termen des Dokuments (nach Stemming und
        Stoppwörter-Entfernung) < minimumNumberOfTerms dann
        Verwirf das aktuell zu bearbeitende Dokument.
        sonst
            wenn Ein ähnliches Patentdokument existiert (String-Vergleich) und das
            Filtern von Patentfamilien-Doppel gewünscht wird dann
            Verwirf das aktuell zu bearbeitende Dokument.
            sonst
                Aktualisiere Dokument- und Kollektionsfrequenz für die Terme des
                aktuell zu bearbeitenden Dokuments.
            Ende
        Ende
    Ende
    für alle Patentdokumente, die aus dem vorherigen Arbeitsschritt entstanden sind,
    tue
        Gewichte die Terme nach dem Okapi bzw. TF-IDF Gewichtungsschema.
        Vervollständige Statistik-Werte (Anzahl Terme, Anzahl Dokumente nur mit
        einem Titel, usw.)
    Ende
    Erstelle die jeweiligen Eingabeformate für die Clustering-Verfahren.
    Schreibe die Statistik in die Statistik-Dateien.
    Initialisiere Datenstrukturen zur Verarbeitung der nächsten Anfrage neu.
Ende
```

Zum Stemming wird der Snowball-Stemmer[1] eingesetzt, der ein regelbasiertes Verfahren zur Abtrennung der Suffixe verwendet. Die zum Stemming eingesetzte Instanz entfernt auch die Stoppwörter, die in Form einer Textdatei im selben Verzeichnis wie die Klasse `PatentPreProcess.java` vorliegen muss. Diese Datei wird als Quelldatei zur Stoppwortlisten-Generierung eingesetzt, wobei pro Zeile ein Stoppwort notiert sein muss.

B.2 ExperimenterGUI

B.2.1 Programmeigenschaften und -fähigkeiten

Die für die Experimente eingesetzte Software zum Clustern arbeitet überwiegend kommandozeilenorientiert (CLUTO, Autoclass, SNN). Nur WEKA und der graphische Aufsatz für CLUTO (gCLUTO) bieten eine graphische Nutzerschnittstelle. Jedoch wurde gCLUTO nicht eingesetzt, da Ergebnisse der Clustereinteilung nur visuell am Bildschirm dargestellt werden konnten, ohne auf den hier vorliegenden Anwendungskontext Rücksicht zu nehmen. D.h., es wurden zwar Cluster angezeigt, jedoch waren die zugehörigen Patentdokumente daraus nicht ersichtlich.

Daher wurde im Zuge der Magisterarbeit eine GUI (Graphical User Interface) entwickelt, die die am Bildschirm eingegebenen Parameter an die kommandozeilenorientierten Programme (in diesem Falle SNN und CLUTO) weiterreicht und ein sofortiges Betrachten der erzeugten Lösung mitsamt des Patentdokumentinhalts ermöglicht. Dies vereinfacht den Umgang mit der Software erheblich, da nicht erst umständlich auf der Kommandozeile lange Pfadangaben zu den Quelldateien und Zieldateien zum Speichern der Ergebnisse angegeben werden müssen.

Die Oberfläche *ExperimenterGUI* (Abbildung B.1) erlaubt

- die Auswahl der Ausgangsdaten, die geclustert werden sollen (über das Dropdown-Feld *input file*),
- die Angabe der Datei, in die das Ergebnis eines Clustering-Laufes geschrieben wird (Textfeld *output file*),
- das Betrachten der Ausgabe der Kommandozeilen-Tools (Feld *Results*)
- die Auswahl, ob das CLUTO bzw. SNN interne Term-Gewichtungsschema benützt werden soll (Sofern ein Dateiname den String „_weighted“ enthält, wird als Standardeinstellung das interne Gewichtungsschema von CLUTO bzw. SNN deaktiviert).
- das Betrachten der erzeugten Cluster

Wenn ein Clustering-Ergebnis angezeigt werden soll (durch Klicken auf Schaltfläche *view*, *autoclass* oder *mainIPC*, wird anhand des im Textfeld *output file* angegebenen

[1] `http://snowball.tartarus.org`, Verifizierungsdatum: 05.10.2004, 10:14 Uhr MEZ

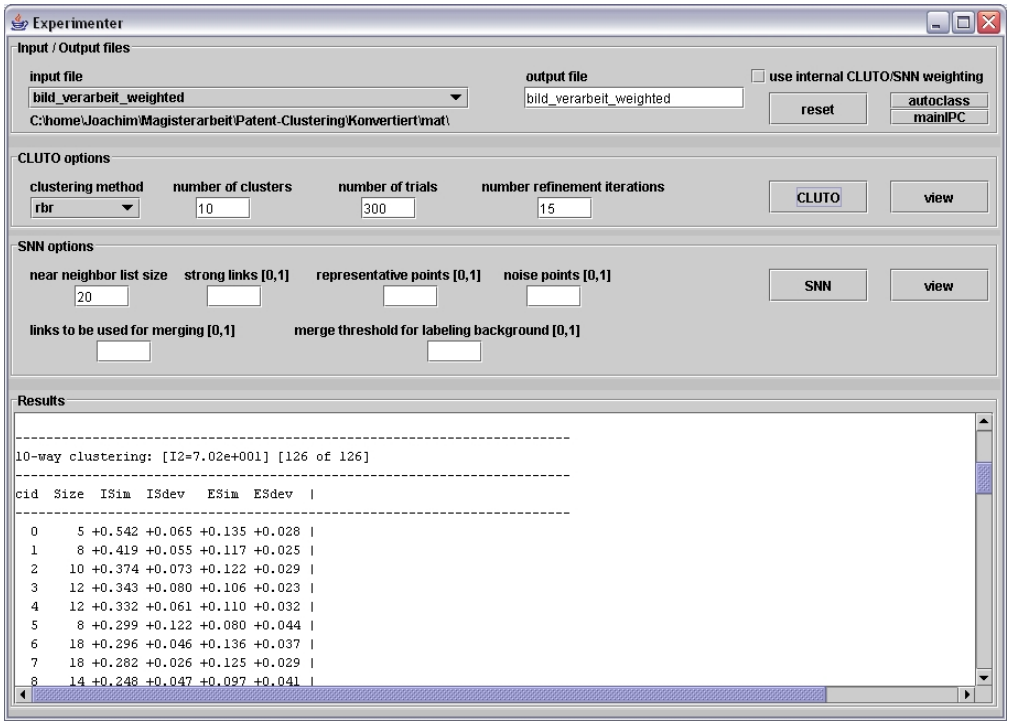


Abbildung B.1: ExperimenterGUI

Dateinamens versucht, diese Ergebnisdatei zu laden. Je nach zur Clusterbildung gewähltem Verfahren, wird diese Datei in dem jeweiligen Verzeichnis ausgehend vom `solutionsDirectory` gesucht.

Die Ergebnisse eines Clustering-Laufes werden in einem separaten Fenster (Abbildung B.2) dargestellt. Im linken Fensterteil werden die Cluster und die zugehörigen Dokumenttitel in einer Baumstruktur angezeigt (alphabetisch sortiert nach Titeln). Der rechte Fensterteil zeigt nach Auswahl eines Patentdokuments den Inhalt des Dokuments an. Insgesamt bietet dieses Fenster zur Ergebnis-Präsentation folgende Funktionen:

- Aufklappen des gesamten Baumes (Schaltfläche *expand all*) bzw. Zusammenklappen des gesamten Baumes (Schaltfläche *collapse all*)
- Farbliche Hervorhebung der Dokumenttitel im linken Fensterteil, die zur selben IPC-Klasse (definiert durch die MainIPC der Patentdokumente) gehören. Diese Funktion lässt sich mittels der Checkbox *highlight same mainIPC* ein- bzw. ausschalten sowie mit nachfolgender Option kombinieren.
- Durch Auswählen der Checkbox *sort by mainIPC* werden die Titel der Dokumente innerhalb jedes Clusters zusätzlich nach ihrer Zugehörigkeit zur selben MainIPC gruppiert.
- Durch Auswahl eines Knotens, der einen Cluster repräsentiert (z.B. *Cluster 1 (6)*), erscheint im rechten Fensterteil eine Übersicht der zum Cluster gehörenden Dokumente und deren Verteilung über die IPC.

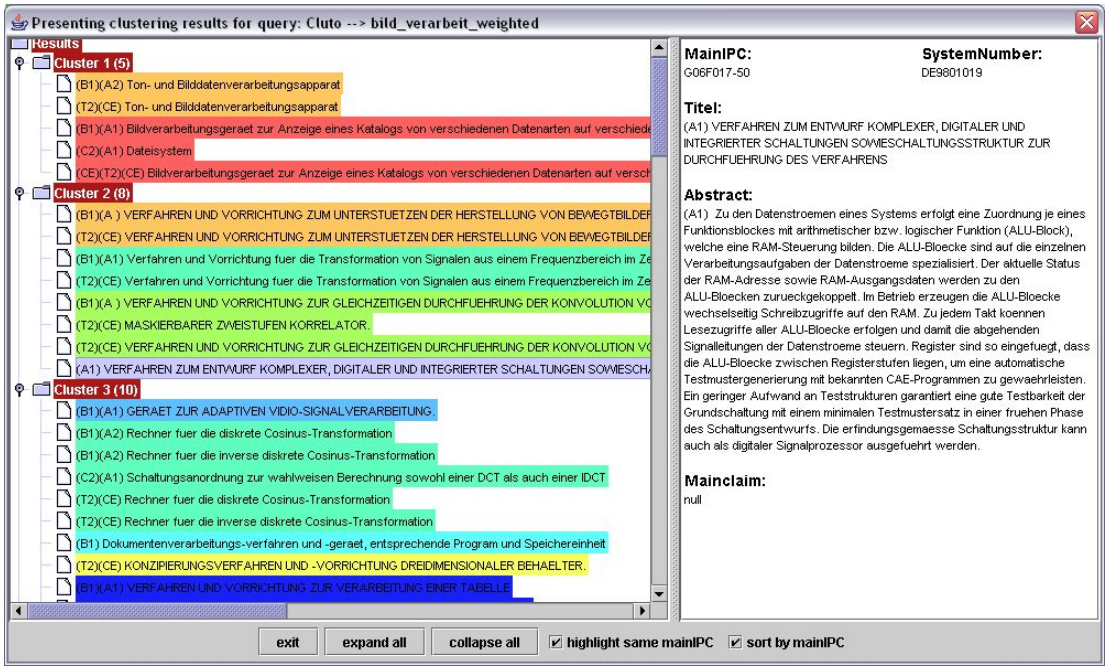


Abbildung B.2: Darstellung eines Resultats eines Clustering-Laufes

B.2.2 Konfiguration

Für den Zugriff auf Dateien, der während des Arbeitens mit dem Tool *Experimenter-GUI* auftritt, muss in einer Konfigurationsdatei `config.ini` der Speicherort einiger Verzeichnisse angegeben werden, so z.B. zur Anzeige der Patentdokumente. Diese Konfigurationsdatei ist im selben Verzeichnis wie die Datei `ExperimenterGUI.java` zu hinterlegen und beinhaltet Angaben zu dem Speicherort der Verzeichnisse,

- ❏ in dem die Patentdokument-Vektoren mit den Inhalten der Dokumente gespeichert sind (`vectorDirectory`), die von der JAVA-Klasse `PatentPrePro-cess.java` erzeugt wurden.
- ❏ das die Dateien mit den SystemNumbers enthält (`clabelDirectory`), die zur korrekten Anzeige der Patentdokumente benötigt werden. Diese Dateien werden bei der Konvertierung mittels der JAVA-Klasse `PatentPrePro-cess.java` im Basisverzeichnis unter `\mat` abgelegt.
- ❏ in dem die Ergebnisse der Clustering-Verfahren gespeichert werden sollen (`solutionsDirectory`).

B.3 Evaluierungstool ClustEv

B.3.1 Programmeigenschaften und -fähigkeiten

Zur Evaluation der Clustering-Lösungen wurde das Programm *ClustEv* im Zuge dieser Magisterarbeit erstellt (`ClustEv.java`). Das Programm dient der Erfassung der

Bewertungen durch Juroren, sowie der automatischen Auszählung der Bewertungen. Bewertet ein Nutzer eine Anfrage, so kann er die Evaluation jederzeit unterbrechen. Seine Bewertungen werden gespeichert und beim Fortfahren wieder geladen. Die Bewertung erfolgt durch Betrachten jedes einzelnen Dokuments eines Clusters und der Entscheidung darüber, ob dieses Dokument in den (Gesamt-)Zusammenhang des Clusters passt oder nicht hinein passt.

Im Rahmen der Auswertung werden zwei Übersichten erzeugt. Zum einen werden getrennt für jeden Juror die Bewertungen einer Anfrage ausgegeben. Zum anderen werden diese Einzelurteile über alle Juroren in einer Gesamtauswertung (pro Anfrage) aufsummiert. In diesen beiden Auswertungen wird für jeden Cluster einer Anfrage die Anzahl der mit „passend“ oder „nicht passend“ bewerteten Dokumente aufgeführt (und eventuell die Anzahl der nicht bewerteten Dokumente), um für jeden Cluster anhand der Summen dieser Werte eine Gesamtwertung wie „passend“, „nicht passend“ oder „unentschieden“ zu erhalten.

B.3.1.1 Hauptfenster

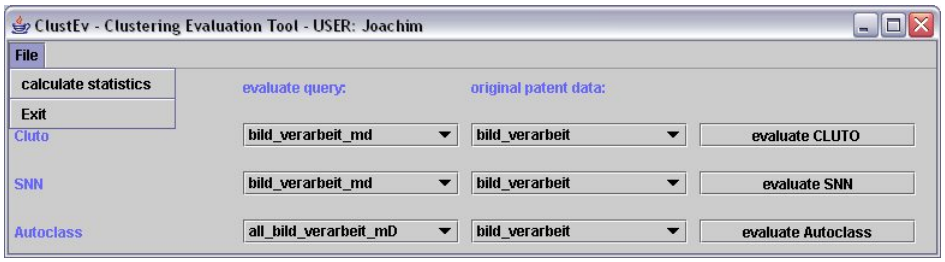


Abbildung B.3: Hauptfenster der Anwendung ClustEv

Im Hauptfenster (Abbildung B.3) muss ein Juror pro Verfahren (in diesem Falle sind dies CLUTO, Autoclass und SNN) mittels Dropdown-Listen die Anfrage (*evaluate query*) und die zugehörige Datenbasis zur Anzeige der Patentdokumente (*original patent data*) auswählen. Durch Klicken auf die *evaluate X* Schaltfläche öffnet sich ein Fenster, in dem die eigentliche Evaluation stattfindet (Abbildung B.4).

B.3.1.2 Abgabe der Bewertungen

Im linken Teil des Fensters (siehe Abbildung B.4) werden die Cluster und die zugehörigen Dokumenttitel in einer Baumstruktur angezeigt (alphabetisch sortiert nach Titeln). Der rechte Fensterteil zeigt nach Auswahl eines Patentdokuments den Inhalt des Dokuments an. Das gesamte Fenster bietet dem Nutzer folgende Funktionalität:

- Aufklappen des gesamten Baumes (Schaltfläche *expand all*) bzw. Zusammenklappen des gesamten Baumes (Schaltfläche *collapse all*)

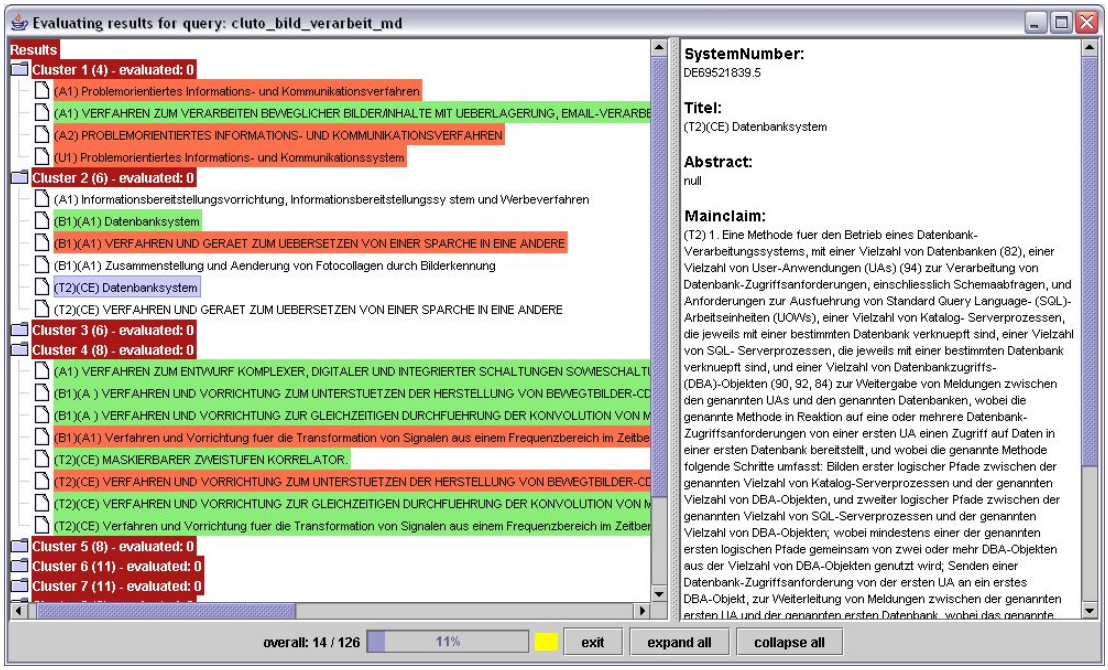


Abbildung B.4: Fenster zur Bewertung einer Anfrage

- Bewertung des aktuell ausgewählten Dokuments mittels Tastendruck:
  Das Dokument passt in den Cluster = Taste **f** (*„document **f**its“*);
  Das Dokument passt nicht in den Cluster = Taste **n** (*„document fits **n**ot“*);
- Fortschrittsanzeige (in der Fußleiste) zum Ablesen der Anzahl der bereits bewerteten Dokumente einer Anfrage.
- Farbliche Hervorhebung des Dokumenttitels, je nach abgegebener Bewertung (rot = Dokument passt nicht in den Cluster; grün = Dokument passt in den Cluster).
- Angabe der Anzahl der bereits bewerteten Dokumente in den Knoten im Baum. Diese Angaben werden erst nach Schließen und erneutem Öffnen aktualisiert. Sie dienen zur Information bei einem Wiedereinstieg in die Bewertung nach einer längeren Bearbeitungspause (mit Schließen des Programms).

B.3.1.3 Auswertung

Durch Auswahl des Menüeintrags *File – calculate statistics* wird das Auswertungsfenster geöffnet (Abbildung B.5). Dieses Fenster lässt sich nur von einem Nutzer öffnen, dessen Nutzername (festgelegt in der Datei `config.ini`) mit dem in der actionPerformed-Methode der Quelltextdatei `PatEvalGUI.java` vorgegebenen Nutzernamen übereinstimmt.

Im oberen Teil des Fensters werden die Dateien angezeigt, die zur Berechnung der Auswertungsergebnisse herangezogen werden. Diese Dateien entstanden aus den Bewertungen der einzelnen Juroren, die im Verzeichnis `storeDirectory` im Rahmen

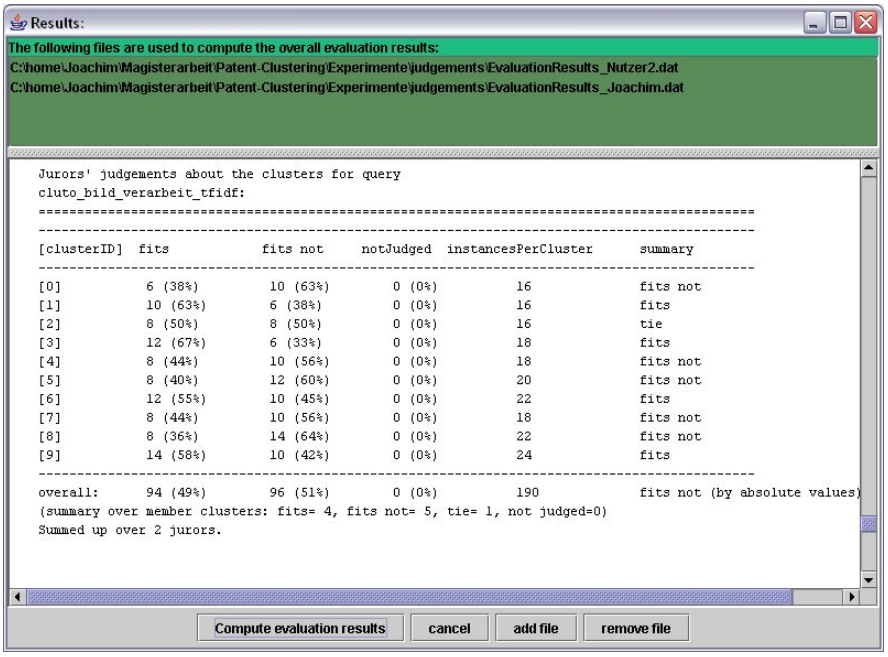


Abbildung B.5: Fenster zur Auswertung der Bewertungen

der Evaluation unter `EvaluationResults_Nutzername` bei jedem Nutzer angefallen sind. Durch Klicken auf die Schaltfläche *add file* können diese Ergebnisdateien mittels eines Datei-Dialoges ausgewählt werden. Soll eine zuvor ausgewählte Datei nicht in die Auswertung einbezogen werden, kann sie im oberen Teil des Dialogfelds selektiert werden und durch Klicken auf die Schaltfläche *remove file* wieder entfernt werden.

Die Auswertung wird durch Klicken auf die Schaltfläche *Compute evaluation results* gestartet. Im unteren Teil des Fensters werden die Ergebnisse dargestellt und zugleich werden im Verzeichnis `storeDirectory` drei CSV-Dateien (Comma Separated Values) angelegt, die zur Weiterverarbeitung z.B. in Microsoft Excel importiert werden können.

B.3.2 Konfiguration

Um mit dem Tool *ClustEv* arbeiten zu können, müssen in einer Konfigurationsdatei `config.ini` Pfadinformationen (Speicherort von bestimmten Dateien) sowie ein Nutzername angegeben werden. Diese Konfigurationsdatei ist im selben Verzeichnis wie die Datei `ClustEv.java` zu hinterlegen und beinhaltet folgende Angaben:

- Verzeichnis, in dem die Patentdokument-Vektoren mit den Inhalten der Dokumente gespeichert sind (`vectorDirectory`). Diese werden durch die JAVA-Klasse `PatentPreProcess.java` erzeugt.
- Verzeichnis, das die Dateien mit den SystemNumbers enthält, die zur korrekten Anzeige der Patentdokumente benötigt werden (`clabelDirectory`). Diese

Dateien werden bei der Konvertierung mittels der JAVA-Klasse `PatentPreProcess.java` im Basisverzeichnis unter `\mat` abgelegt.

- Basisverzeichnis der Lösungen (`solutionsDirectory`), in dem die zu evaluierenden Lösungen, je nach verwendetem Verfahren, in Unterverzeichnissen gespeichert sind. Beispielsweise müssen sich sämtliche Anfragen, die vom Verfahren CLUTO erstellt wurden, im Verzeichnis (`\solutionsDirectory\CLUTO`) befinden.
- Verzeichnis, in dem die Bewertungen gespeichert werden (`storeDirectory`).
- Nutzername zur Identifikation der Evaluationsergebnisse bei der Auswertung (`userName`)

Eigenständigkeitserklärung

Ich erkläre, dass ich diese Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die diesen Quellen und Hilfsmitteln wörtlich oder sinngemäß entnommenen Ausführungen als solche kenntlich gemacht habe.

Hildesheim, den 22. November 2004 *Joachim Pfister*